ReflexFlow: Rethinking Learning Objective for Exposure Bias Alleviation in Flow Matching

Guanbo Huang[1,*], **Jingjia Mao**[1,*], **Fanding Huang**[1,*], **Fengkai Liu**[1], **Xiangyang Luo**[1], **Yaoyuan Liang**[1], **Jiasheng Lu**[2], **Xiaoe Wang**[2], **Pei Liu**[2], **Ruiliu Fu**[2,†], **Shao-Lun Huang**[1,†]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Media Technology Lab, Huawei, China

Abstract

*Despite tremendous recent progress, Flow Matching methods still suffer from exposure bias due to discrepancies in training and inference. This paper investigates the root causes of exposure bias in Flow Matching, including: (1) the model lacks generalization to biased inputs during training, and (2) insufficient low-frequency content captured during early denoising, leading to accumulated bias. Based on these insights, we propose **ReflexFlow**, a simple and effective reflexive refinement of the Flow Matching learning objective that dynamically corrects exposure bias. ReflexFlow consists of two components: (1) **Anti-Drift Rectification (ADR)**, which reflexively adjusts prediction targets for biased inputs utilizing a redesigned loss under training-time scheduled sampling; and (2) **Frequency Compensation (FC)**, which reflects on missing low-frequency components and compensates them by reweighting the loss using exposure bias. ReflexFlow is model-agnostic, compatible with all Flow Matching frameworks, and improves generation quality across datasets. Experiments on CIFAR-10, CelebA-64, and ImageNet-256 show that ReflexFlow outperforms prior approaches in mitigating exposure bias, achieving a 35.65% reduction in FID on CelebA-64. Code will be made available in* https://github.com/wuliwuliy/ReflexFlow.git.

1 Introduction

Recently, Flow Matching (FM) [2, 26, 27] has emerged as a promising framework for generative modeling. By modeling continuous-time flows, FM offers a more direct and efficient training objective compared to traditional diffusion-based approaches [18, 43, 44]. Its strong empirical performance and theoretical simplicity have established FM as a foundational method across diverse generation tasks, including image, video, and audio generation tasks [5, 9, 23, 30, 46, 48].

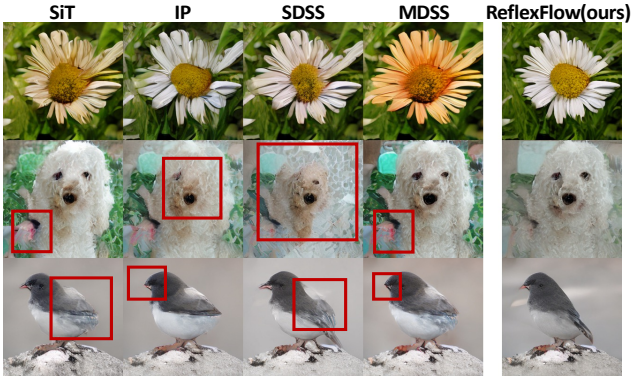

Figure 1. **Qualitative Comparison of Generated Samples.** All models are trained from scratch for 500K iterations using the SiT-B/4 backbone on ImageNet-256. Red boxes highlight visual artifacts produced by competing methods (SiT [30], IP [31], SDSS [37], and MDSS [37]). ReflexFlow produces the most faithful and visually realistic results.

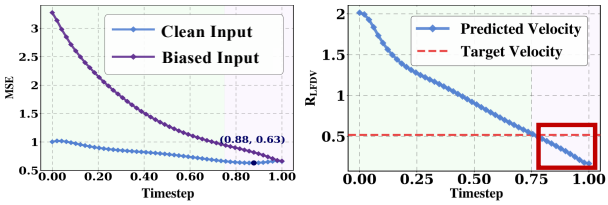

Figure 2. **Quantitative Analysis of Exposure Bias in Flow Matching. Left:** Mean squared error (MSE) of model outputs under clean inputs (forward-perturbed samples) versus biased inputs (reverse-predicted samples), showing that exposure bias significantly increases the MSE. **Right:** Frequency distribution of model outputs compared to ground truth velocity under clean inputs. Early denoising (highlighted by the purple background) predictions lack low-frequency components (marked by the red box), while later timesteps (highlighted by the green background) show a deficiency in high-frequency components.

However, FM still suffers from exposure bias caused by mismatches between training and inference inputs. As shown in Fig. 2 (left), the forward process having clean in-

[*]Equal contribution.
[†]Corresponding authors.

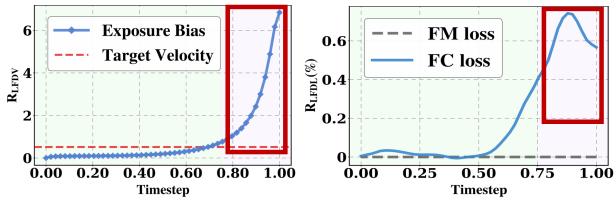

Figure 3. **Exposure Bias and Frequency Compensation in Flow Matching. Left:** Evolution of the frequency distribution of exposure bias across timesteps, showing that early timesteps are dominated by low-frequency components (marked by the red box), while later timesteps are dominated by high-frequency components. **Right:** Compared to the original learning objective, the exposure-bias-reweighted loss emphasizes low-frequency components more at early timesteps (marked by the red box), demonstrating its ability to compensate for frequency discrepancy.

put produces perturbed samples along the ideal flow path, while the reverse process having biased input yields noisy predicted samples that drift away from it. While several studies have investigated this issue within the DDPM framework [11, 24, 25, 32, 52], their solutions remain largely heuristic. Some approaches introduce inference-time regularization without additional training [24, 32, 52], while others aim to mitigate bias during training by simulating the inference process. For instance, IP [31] injects perturbations into real data during training, and [37] proposes Multi-step Denoising Scheduled Sampling (MDSS) and Single-step Denoising Scheduled Sampling (SDSS), aligning the training target with the anticipated inference dynamics. Despite improving the robustness of the model to biased inputs, these methods do not fundamentally eliminate the inherent bias induced by the training-inference mismatch.

Our work identifies the fundamental sources of exposure bias in Flow Matching and analyzes their underlying mechanisms. First, the model is trained exclusively on clean inputs, which undermines its robustness to the input distortions that naturally emerge during multi-step inference. This discrepancy is clearly demonstrated in Fig. 2 (left), where the model's predictions diverge significantly between clean and biased inputs. Prior methods inject biased inputs during training to improve robustness, yet lack a principled objective for such inputs, leading to unmitigated bias. Second, frequency analysis of the model's early denoising predictions uncovers a key limitation, a significant lack of low-frequency components in the initial outputs. This frequency deficiency results in substantial error propagation during early denoising stages. To quantify this, we introduce two metrics. The R_{LFDV} measures the ratio of low- to high-frequency energy in the model's predictions, capturing how much low-frequency content the model produces at each step. The R_{LFDL} measures the relative emphasis the learning objective places on low- versus high-frequency regions of the target velocity. Detailed definitions are provided in Sec. 4.2. As highlighted in the red box of Fig. 2 (right), the low-frequency energy in FM predictions remains consistently below that of the ground truth during early denoising, showing that the model insufficiently captures low-frequency components. Since these components encode the most critical structural information for generation, their insufficient representation significantly degrades overall generation quality. Interestingly, we find that exposure bias exhibits a complementary, low-frequency–dominated property at early timesteps, as shown in Fig. 3 (left).

Based on these findings, we propose **ReflexFlow**, which redesigns the Flow Matching learning objective to reflexively correct its own bias during both training and training-time inference, jointly mitigating exposure bias. ReflexFlow consists of two core components: *(i)* **Anti-Drift Rectification (ADR)**: applied during training-time inference, it mitigates prediction drift by reconstructing a learning objective that explicitly points toward the true data distribution, guiding the model back to the ideal flow path. *(ii)* **Frequency Compensation (FC)**: applied in the original training objective, it leverages exposure bias itself as a reweighting signal to enhance the learning of missing frequency components. As shown in Fig. 3 (right), the reweighted loss emphasizes absent low-frequency information, mitigating exposure bias at its root. The framework is simple, model-agnostic and effective. Fig. 1 highlights the qualitative improvements achieved by ReflexFlow.

Our main contributions are summarized as follows:

- We present the first systematic study of exposure bias in Flow Matching, analyze its root causes and propose ADR — a theoretically grounded method that directly mitigates prediction drift.
- We observe that exposure bias inherently supplements missing low-frequency information and leverage this property to design FC. It uses negative feedback to reinforce the learning of deficient frequency components, improving the original learning objective.
- We integrate these two submethods into a unified framework, ReflexFlow, and conduct extensive experiments on CIFAR-10, ImageNet-256 and CelebA-64 datasets. Empirical results show that ReflexFlow consistently improves generation quality, mitigates exposure bias and achieves state-of-the-art performance.

2 Related Work

Exposure Bias. Exposure bias arises from the mismatch between training and inference inputs in sequential models [4, 36, 41, 49, 56, 57]. Early approaches such as DAD [47] combines ground truth and predicted tokens during training, and SS [8] samples biased inputs to better approximate inference. In diffusion models, DDPM-IP [31] first formalizes exposure bias and perturbs training samples to simulate it. EP-DDPM [25] studies error propagation and introduces a cumulative-error regularizer. AE-DDPM [55] mit-

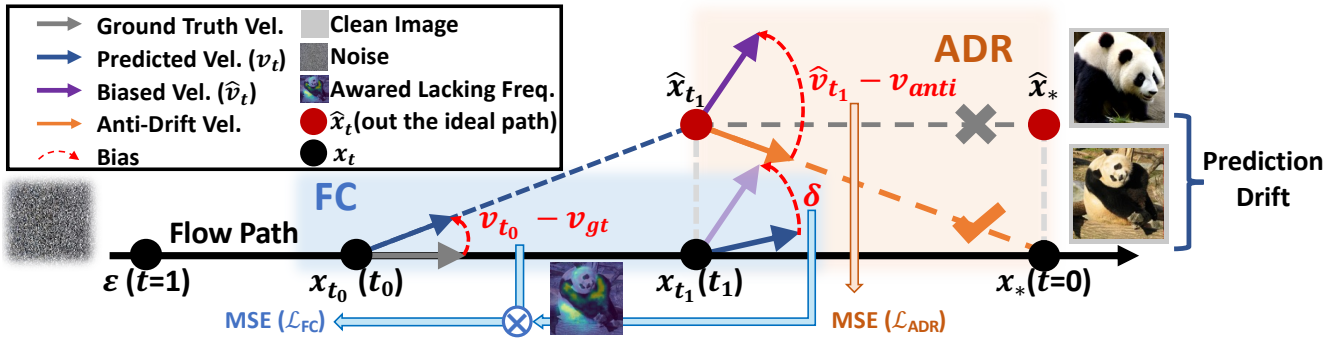

Figure 4. **Overview of ReflexFlow.** (a) **Anti-Drift Rectification**: We construct a new learning objective that guides the model from the drifted intermediate distribution toward the true data distribution, enabling effective correction of prediction drift. (b) **Frequency Compensation**: To mitigate the low-frequency deficiency at HFT, we use exposure bias as a negative-feedback signal to reweight the original objective. The final objective jointly optimizes the weighted original objective and rectification objective.

igates exposure bias via prompt learning, and MCDO [53] imposes manifold constraints. Training-free techniques include a time-shift sampler (TS-DDPM [24]), noise scaling (ES-DDPM [32]), leveraging training-distribution leak signals [11] and frequency-domain analysis with wavelet-based regularization [52]. Among prior works, MDSS [37] is most relevant to our work: it simulates multi-step inference during training and aligns predictions with the original ground truth. However, within the FM framework, direct alignment to ground truth remains insufficient. We provide the first systematic study of exposure bias in FM and introduce a new reflexive learning objective to address it.

Reweighting loss for emphasizing saliency area alignment. Several works explore reweighting strategies in loss functions to emphasize salient or informative regions, guiding models to focus on the most influential areas. In decomposable or multi-stage generation, Decomposable Flow Matching [16] applies spatially varying masks to highlight critical regions on outputs. In video generation, Proteus-ID [54] and MotiF [50] adopt motion-aware weighting to prioritize frequently moving or semantically important regions. MotionCharacter [12] further improves learning efficiency by emphasizing motion-sensitive areas. Beyond spatial analysis, Latent Wavelet Diffusion [42] performs frequency analysis and reweights loss to concentrate learning on frequency dominant components. Together, these methods demonstrate that adaptive loss reweighting can effectively direct model capacity toward salient regions, improving robustness and representation efficiency.

3 Preliminaries

3.1 Flow Matching

Flow Matching (FM) formulates generative modeling as learning a velocity field that transports a simple prior $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to the target data distribution $\mathbf{x}_* \sim p_{\text{data}}(\mathbf{x})$. A sample at time t is constructed as

$$\mathbf{x}_t = a_t \mathbf{x}_* + b_t \boldsymbol{\epsilon}, \tag{1}$$

where a_t and b_t are predefined schedules, typically $a_t = 1 - t$ and $b_t = t$. The corresponding velocity is $\mathbf{v}_t = \frac{da_t}{dt}\mathbf{x}_* + \frac{db_t}{dt}\boldsymbol{\epsilon} = \boldsymbol{\epsilon} - \mathbf{x}_*$, which remains constant along the linear path. FM trains a neural network $\mathbf{v}_\theta(\mathbf{x}, t)$ to approximate this conditional velocity by minimizing

$$\mathcal{L}_{\text{FM}}(\theta) := \mathbb{E}_{\mathbf{x}_*, \boldsymbol{\epsilon}, t}\left[\left\|\mathbf{v}_\theta(\mathbf{x}_t, t) - (\boldsymbol{\epsilon} - \mathbf{x}_*)\right\|^2\right]. \tag{2}$$

This objective encourages the model to learn a time-dependent velocity field that consistently transports intermediate distribution towards the true data distribution.

3.2 Fourier Frequency Analysis

The Fourier transform provides a way to represent spatial-domain signals in the frequency domain, decomposing them into sinusoidal components of different frequencies and amplitudes. Since the predicted velocity field $\mathbf{v} \in \mathbb{R}^{H \times W}$ shares a similar spatial structure with images, we can directly apply frequency-domain analysis to it. The 2D Discrete Fourier Transform (DFT) of $\mathbf{v}$ is defined as:

$$\mathbf{V}(u, v) = \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} \mathbf{v}(x, y)\, e^{-i2\pi\left(\frac{ux}{H} + \frac{vy}{W}\right)}, \tag{3}$$

where (u, v) denotes the frequency coordinates, and $i = \sqrt{-1}$ is the imaginary unit. The resulting $\mathbf{V}(u, v)$ consists of a real part $R(\mathbf{V})$ and an imaginary part $I(\mathbf{V})$, from which the amplitude and phase spectra can be derived as:

$$\begin{aligned} |\mathbf{V}(u, v)| &= \sqrt{R(\mathbf{V})^2 + I(\mathbf{V})^2}, \\ \angle\mathbf{V}(u, v) &= \arctan\left(\frac{I(\mathbf{V})}{R(\mathbf{V})}\right). \end{aligned} \tag{4}$$

The inverse Fourier transform reconstructs the spatial-domain signal from its frequency representation:

$$\mathbf{v}(x,y) = \frac{1}{HW}\sum_{u=0}^{H-1}\sum_{v=0}^{W-1}\mathbf{V}(u,v)\, e^{i2\pi\left(\frac{ux}{H}+\frac{vy}{W}\right)}. \quad (5)$$

In practice, we adopt the Fast Fourier Transform (FFT) and its Inverse (IFFT) for efficient computation.

3.3 Exposure Bias Problem Formulation

In this section, we define the exposure bias problem that arises in the FM framework. Specifically, during training, the model $\mathbf{v}_\theta(\mathbf{x}_t, t)$ receives input $\mathbf{x}_t$ obtained from a linear interpolation between the true data distribution $\mathbf{x}_*$ and Gaussian noise $\boldsymbol{\epsilon}$, as defined in Eq. 1. During inference, however, the input $\hat{\mathbf{x}}_t$ is generated from the previous prediction $\hat{\mathbf{v}}_{t-1}$, which inevitably accumulates prediction errors. This mismatch leads to a drift between $\mathbf{v}_\theta(\mathbf{x}_t, t)$ in training and $\mathbf{v}_\theta(\hat{\mathbf{x}}_t, t)$ in inference, and the errors propagate through timesteps, progressively degrading sample quality.

To better characterize this training-testing discrepancy, we extend the definition of exposure bias to our setting. In our training strategy, one-step inference is explicitly simulated, as illustrated in Fig. 4. At time t_0, the model takes the ideal perturbed input $\mathbf{x}_{t_0}$ and predicts the velocity $\mathbf{v}_\theta(\mathbf{x}_{t_0}, t_0)$. Due to inevitable prediction errors, the estimated velocity deviates from the ground-truth direction $\boldsymbol{\epsilon} - \mathbf{x}_*$. The model then proceeds to the next timestep t_1, producing $\hat{\mathbf{x}}_{t_1} = \mathbf{x}_{t_0} + (t_1 - t_0)\cdot\mathbf{v}_\theta(\mathbf{x}_{t_0}, t_0)$, which diverges from the perturbed sample along the ideal flow trajectory, thereby introducing prediction drift. This prediction drift further propagates, as the subsequent velocity estimate $\mathbf{v}_\theta(\hat{\mathbf{x}}_{t_1}, t_1)$ inherently accumulates the previous deviation into the next prediction.

To formalize exposure bias at a single timestep, consider the model output at t_1 without being affected by exposure bias would be $\mathbf{v}_\theta(\mathbf{x}_{t_1}, t_1)$. Consequently, we define the exposure bias for any timestep t as:

$$\boldsymbol{\delta}_{\text{exp},t} = \mathbf{v}_\theta(\hat{\mathbf{x}}_t, t) - \mathbf{v}_\theta(\mathbf{x}_t, t). \quad (6)$$

This quantity represents the change in the predicted velocity caused solely by replacing the ideal input $\mathbf{x}_t$ with the biased input $\hat{\mathbf{x}}_t$.

4 Method

This section presents our method, *ReflexFlow*, which consists of two core components as shown in Fig. 4: *Anti-Drift Rectification* (ADR) and *Frequency Compensation* (FC).

4.1 Anti-Drift Rectification (ADR): A Reflexive Objective with Theoretical Guarantees

Following the problem formulation in Sec. 3.3, to analyze the multi-step effect of exposure bias, we first derive a formal bound on the accumulation error at the final timestep of FM original loss.

As we mainly analyze the reverse denoising process rather than the forward perturbation process, we reparameterize the time by defining $\tau = 1-t$, which increases from 0 to 1 along the denoising trajectory. Let $\mathbf{e}_{\tau_k} := \hat{\mathbf{x}}_{\tau_k} - \mathbf{x}_{\tau_k}$ be the potential error at the k-th timestep between the drifted-sampling path $\hat{\mathbf{x}}_{\tau_k}$ and the ideal interpolation path $\mathbf{x}_{\tau_k}$ defined in Eq. 1, and the final timestep is K at the predicted endpoint $\hat{\mathbf{x}}_*$. We define the residual relative to ground truth velocity as $\mathbf{r}_{\tau_k} := \mathbf{v}_\theta(\mathbf{x}_{\tau_k}, \tau_k) - (\boldsymbol{\epsilon} - \mathbf{x}_*)$, which is the prediction error of the model on the ideal path, and the exposure bias $\boldsymbol{\delta}_{\text{exp},\tau_k}$ is defined in Eq. 6.

Proposition 4.1 (Error Bounded by FM Target)**.** *The bound for the expected final sampling error $\mathbb{E}\|\mathbf{e}_{\tau_k}\|$ under the FM objective can be expressed as the sum of the integrated ground truth velocity residual and the integrated exposure bias (more details in supplementary materials), which is formulated as follows:*

$$\mathbb{E}\|\mathbf{e}_{\tau_k}\| \le \underbrace{\int_0^1 \sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2}d\tau}_{\textit{Integrated GT Velocity Residual}} + \underbrace{\int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{exp,\tau}\|^2}d\tau}_{\textit{Integrated Exposure Bias}} \quad (7)$$

Remark 4.2. In Eq. 7, the bound on expected accumulation error decomposes into two terms: the *Integrated Ground Truth Velocity Residual* and the *Integrated Exposure Bias*. Standard FM training (Eq. 2) only minimizes the first term ($\mathbb{E}\|\mathbf{r}_\tau\|^2$). This leaves the exposure bias unaddressed, allowing it to accumulate and contribute linearly to the overall accumulation error.

This finding provides the theoretical foundation for our ADR approach. Since the original objective is blind to the exposure-bias term, we must introduce a training signal that explicitly constrains this drift. However, directly using exposure bias as a loss is ineffective, as it encourages the model to follow the drifted path, where the original ground-truth velocity can no longer guide it back to the target.

To guide the deviated $\hat{\mathbf{x}}_t$ back to the ideal flow path during training, we align the model's output velocity $\mathbf{v}_\theta(\hat{\mathbf{x}}_t, t)$ at the predicted intermediate sample, with the velocity pointing from $\hat{\mathbf{x}}_t$ to the target endpoint $\mathbf{x}_*$. This anti-drift velocity target is defined analogously to the standard flow matching objective:

$$\mathbf{v}_{\text{ADR},t}(\hat{\mathbf{x}}_t \mid \mathbf{x}_*) = \hat{\mathbf{x}}_t - \mathbf{x}_*. \quad (8)$$

Considering the discrepancy in magnitude between the predicted and anti-drift velocities, we focus on aligning their directions. The alignment is implemented by minimizing the following anti-drift rectification loss, which is a regularization term added on the original FM loss:

$$\mathcal{L}_{\text{ADR}} := \mathbb{E}_{\mathbf{x}_*,\hat{\mathbf{x}}_t,t}\left[\left\| \frac{\mathbf{v}_\theta(\hat{\mathbf{x}}_t,t)}{\|\mathbf{v}_\theta(\hat{\mathbf{x}}_t,t)\|_2} - \frac{\mathbf{v}_{\text{ADR},t}(\hat{\mathbf{x}}_t \mid \mathbf{x}_*)}{\|\mathbf{v}_{\text{ADR},t}(\hat{\mathbf{x}}_t \mid \mathbf{x}_*)\|_2} \right\|^2\right]. \tag{9}$$

Proposition 4.3 (Error Bounded by ADR)**.** *The bound form with ADR of expected final sampling error $\mathbb{E}\|\mathbf{e}_{\tau_k}\|$ is governed by the integrated ADR residual and a constant path-geometry term (more details in supplementary materials):*

$$\mathbb{E}\|\mathbf{e}_{\tau_K}\| \le \underbrace{\int_0^1 e^{1-\tau}\sqrt{\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2}d\tau}_{\textit{Integrated ADR Residual}} + \underbrace{(e-2)\|\boldsymbol{\epsilon}-\mathbf{x}_*\|}_{\textit{Path-Geometry Constant}}, \tag{10}$$

where $\boldsymbol{\phi}_\tau := \mathbf{v}_\theta(\hat{\mathbf{x}}_\tau,\tau) - (\hat{\mathbf{x}}_\tau - \mathbf{x}_*)$ is the ADR residual. *This term represents the velocity error on the biased $\hat{\mathbf{x}}_\tau$ relative to the target $(\hat{\mathbf{x}}_\tau - \mathbf{x}_*)$, which we introduce in Eq. 8.*

Remark 4.4. Eq. 10 differs fundamentally from Eq. 7. Here, the exposure bias term $\boldsymbol{\delta}_{\text{exp},\tau}$ is algebraically cancelled, and the remaining *Path-Geometry Constant* is independent of model errors. Thus, the final sampling error is determined not by uncontrolled bias but only by the *Integrated ADR Residual* $\boldsymbol{\phi}_\tau$, which is exactly what ADR aims to minimize, as implemented in Eq. 9. In this way, ADR turns the uncontrolled exposure bias into a learnable quantity that the model explicitly reduces.

Lemma 4.5 (Coupling between FM and ADR residuals)**.** *For any $\tau \in [0,1]$, the residuals satisfy the exact identity:*

$$\boldsymbol{\phi}_\tau = \mathbf{r}_\tau + \boldsymbol{\delta}_{\text{exp},\tau} + (\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau). \tag{11}$$

Corollary 4.6 (Order-equivalence of FM and ADR bounds)**.** *Let B_{FM} and B_{ADR} denote the error bounds in Propositions 4.1 and 4.3, respectively, i.e.,*

$$B_{\text{FM}} := \int_0^1 \sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2}\,d\tau + \int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau}\|^2}\,d\tau, \tag{12}$$

$$B_{\text{ADR}} := \int_0^1 e^{1-\tau}\sqrt{\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2}\,d\tau + (e-2)\,\|\boldsymbol{\epsilon}-\mathbf{x}_*\|. \tag{13}$$

Then there exist absolute constants $C_1, C_2 > 0$, independent of $\mathbf{v}_\theta$, such that:

$$B_{\text{ADR}} \;\le\; C_1\, B_{\text{FM}} + C_2\,\|\boldsymbol{\epsilon}-\mathbf{x}_*\|. \tag{14}$$

Remark 4.7. When B_{ADR} and B_{FM} are not of the same magnitude order, it becomes unclear how to compare their effectiveness in measuring the sampling error. Corollary B.2 shows that, even in the worst case, the B_{ADR} remains of the same order as the original B_{FM}. This ensures that B_{ADR} provides a estimate of sampling error with reliability comparable to B_{FM}. Unlike B_{FM}, however, B_{ADR} also includes the exposure bias term that FM training does not minimize. As a result, the ADR objective offers more complete control over the sources of sampling error and can more effectively alleviate exposure bias.

4.2 Frequency Analysis Metrics

To provide a more comprehensive understanding of the frequency characteristics of velocity, we introduce two new metrics: the Low-Frequency Dominant Ratio on Velocity (R_{LFDV}), which quantifies the dominance of low-frequency energy in velocity maps, and the Low-Frequency Dominant Ratio on Loss (R_{LFDL}), which measures the relative emphasis of the loss on low-frequency components over high-frequency ones with respect to the ground-truth velocity maps.

We first introduce the computation of R_{LFDV}. Specifically, given the predicted velocity $\mathbf{v} \in \mathbb{R}^{H\times W}$ produced by the FM model, we perform a 2D FFT on it:

$$\mathbf{V}(u,v) = \text{FFT}(\mathbf{v}(i,j)). \tag{15}$$

To investigate the proportion of low- and high-frequency bands, we apply Low-Pass Filter ($\mathbf{F}_{LP}$) and High-Pass Filter ($\mathbf{F}_{HP}$) defined as:

$$\begin{aligned} \mathbf{F}_{\text{LP}}(u,v) &= \mathbb{I}\left[D(u,v) \le D_{\text{cutoff}}\right], \\ \mathbf{F}_{\text{HP}}(u,v) &= \mathbf{1} - \mathbf{F}_{\text{LP}}(u,v), \end{aligned} \tag{16}$$

where $D(u,v) = \sqrt{u^2+v^2}$ is the frequency magnitude and D_{cutoff} is the cutoff frequency.

We compute the energy ratio between the low- and high-frequency components and defines R_{LFDV} as follows:

$$R_{\text{LFDV}} := \frac{\sum_{u,v} ||\mathbf{V}(u,v)||^2 \cdot \mathbf{F}_{\text{LP}}(u,v)}{\sum_{u,v} ||\mathbf{V}(u,v)||^2 \cdot \mathbf{F}_{\text{HP}}(u,v)}. \tag{17}$$

We then define the R_{LFDL}. Here, we consider the target velocity map $\mathbf{v}_{\text{target}} \in \mathbb{R}^{H\times W}$ and the MSE loss map $\boldsymbol{\mathcal{L}} \in \mathbb{R}^{H\times W}$. To compute it, we divide $\mathbf{v}_{\text{target}}$ into high- and low-frequency regions. The magnitude of the loss map $\boldsymbol{\mathcal{L}}$ in each region reflects the relative emphasis of loss. Thus, we separately sum the loss values within the low- and high-frequency regions to quantify how the loss distributes its emphasis across different frequency components. This metric further serves as an indicator of the capability of loss to compensate for frequency discrepancies.

We first apply Eq. 15 and 16 to perform the FFT of $\mathbf{v}_{\text{target}}$ and compute the corresponding low- and high-frequency masks in the frequency domain. These masks are then used to separate $\mathbf{v}_{\text{target}}$ into its low- and high-frequency components, which are subsequently transformed back to the spatial domain via IFFT. Finally, we compute the salient low- and high-frequency region masks in the spatial domain by thresholding the corresponding values according to their percentile distributions.

$$\begin{aligned} \mathbf{v}_{\text{low}} &= \text{IFFT}(\mathbf{V}\cdot\mathbf{F}_{\text{LP}}), \\ \mathbf{v}_{\text{high}} &= \text{IFFT}(\mathbf{V}\cdot\mathbf{F}_{\text{HP}}), \\ \mathbf{M}_{\text{LFR}}(i,j) &= \mathbb{I}\big[\mathbf{v}_{\text{low}} > p(\mathbf{v}_{\text{low}})\big], \\ \mathbf{M}_{\text{HFR}}(i,j) &= \mathbb{I}\big[\mathbf{v}_{\text{high}} > p(\mathbf{v}_{\text{high}})\big], \end{aligned} \tag{18}$$

where $p(\cdot)$ denotes a percentile threshold function, $\mathbf{v}_{\text{low}}$ and $\mathbf{v}_{\text{high}}$ represents the low- and high-frequency components of $\mathbf{v}_{\text{target}}$. The resulting masks, $\mathbf{M}_{\text{LFR}}$ and $\mathbf{M}_{\text{HFR}}$, indicate the low- and high-frequency region masks, respectively.

Finally, we compute R_{LFDL} as follows:

$$R_{\text{LFDL}} = \frac{\sum \tilde{\boldsymbol{\mathcal{L}}} \cdot \mathbf{M}_{\text{LFR}}}{\sum \tilde{\boldsymbol{\mathcal{L}}} \cdot \mathbf{M}_{\text{HFR}}}, \quad \text{where} \quad \tilde{\boldsymbol{\mathcal{L}}}^{(i,j)} = \frac{\boldsymbol{\mathcal{L}}^{(i,j)}}{\sum_{i=1}^{H}\sum_{j=1}^{W} \boldsymbol{\mathcal{L}}^{(i,j)}}. \tag{19}$$

The trend of R_{LFDL} in Fig. 3 (right) reveals how the loss distributes attention across low- and high- frequency regions during training.

4.3 Frequency Compensation (FC)

Using the two metrics proposed in Sec. 4.2, we analyze exposure bias during training and find that prediction drift primarily arises from frequency deficiency. As shown in Fig. 2 (right), during the timestep t decreasing, the prediction of model initially contain more high-frequency components than $\mathbf{v}_{\text{target}}$, and later contain more low-frequency components. Accordingly, the training process can be divided into two phases based on the dominant frequency components: High-Frequency Timesteps (HFT), during which low-frequency information needs to be compensated, and Low-Frequency Timesteps (LFT), during which high-frequency information needs to be compensated. Notably, exposure bias exhibits a complementary property, as illustrated in Fig. 3 (left).

Motivated by this, we visualize the spatial distribution of exposure bias. As shown in Fig. 5, we extract the dominant low- and high-frequency components of the original image as references. The heatmaps of exposure bias show strong concentration in low-frequency regions during HFT, almost fully overlapping with the low-pass results of the original image. This indicates that exposure bias naturally highlights low-frequency content of the samples. Inspired by [12, 42, 50, 54], which demonstrate that reweighting the loss can enhance saliency alignment, we design a negative-feedback weight mask for the original loss based on exposure bias. The weight is defined as:

$$\mathbf{W}_{\text{exp},t}^{(i,j)} = 1 + \alpha \frac{\boldsymbol{\delta}_{\text{exp},t}^{(i,j)}}{\sum_{i=1}^{H}\sum_{j=1}^{W} \boldsymbol{\delta}_{\text{exp},t}^{(i,j)}}, \tag{20}$$

where α is a scaling coefficient. We then use $\mathbf{W}_{\text{exp},t}$ to reweight the original FM learning objective:

$$\mathcal{L}_{\text{FC}} := \mathbb{E}_{\mathbf{x}_*, \boldsymbol{\epsilon}, t}\left[\left\| \mathbf{W}_{\text{exp},t} \cdot \left(\mathbf{v}_{\theta,t} - (\boldsymbol{\epsilon} - \mathbf{x}_*)\right)\right\|^2\right]. \tag{21}$$

This weighting strategy adaptively adjusts the loss distribution according to the exposure bias, emphasizing the missing frequency components at each timestep.

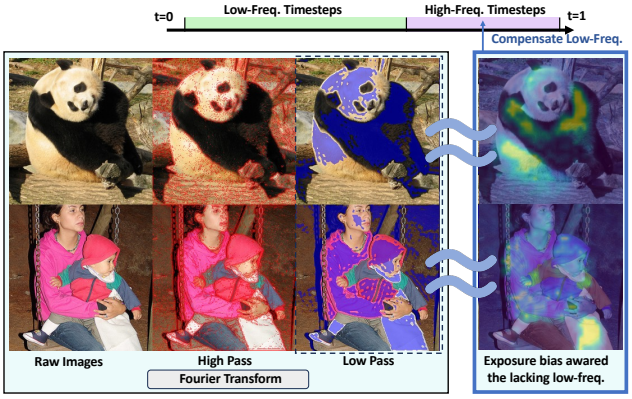

Figure 5. **Visualization of dominant low- and high-frequency regions of the original images (left) and the corresponding exposure bias heatmaps (right) at HFT**. The exposure bias heatmaps are compared with the low-(blue mask) and high-pass(red mask) filtered versions of the ground-truth images, revealing that the regions emphasized by exposure bias correspond closely to the low-frequency components. Additional examples are provided in supplementary materials.

By analyzing $\mathcal{L}_{\text{FC}}$ using the R_{LFDL}, as illustrated in Fig. 3 (right), we find that the exposure-bias-weighted loss effectively emphasizes low-frequency information during the HFT phase. Since these early timesteps are crucial for the quality of the generation of model [3, 20, 51, 58], this emphasis for compensation can significantly improve the final quality of the model generation.

4.4 ReflexFlow

ADR learns to correct the exposure bias that has already occurred during the simulated inference phase, while FC mitigates exposure bias at its source during training. These two mechanisms complement each other across the two stages. We integrate them into a unified framework, **ReflexFlow**, which reflexively adjusts learning objectives to jointly mitigate exposure bias in FM. The final learning objective combines the two mechanisms as follows:

$$\mathcal{L} := \beta_1 \mathcal{L}_{\text{ADR}} + \beta_2 \mathcal{L}_{\text{FC}}, \tag{22}$$

where β_1 and β_2 are weighting coefficients for ADR and FC, respectively. ReflexFlow thus provides a simple yet effective solution to exposure bias, enhancing both prediction accuracy and generation quality. The training procedure of ReflexFlow is summarized in Algorithm 1.

5 Experiments

In this section, we provide the experimental details and compare the performance of different methods.

5.1 Experiment Setting

Main Implementation. We evaluate our methods on conditional image generation tasks on ImageNet-256 [7] and

Table 1. **Quantitative comparison of different methods across different datasets.** Class-conditional (without / with cfg=1.5) generation on ImageNet-256 and CIFAR-10, and unconditional generation on CIFAR-10 and CelebA-64. All entries are reported using 50-NFE samplings. Red highlights improvements by our method, green indicates deficits compared to gray baseline. Models of CIFAR-10 and CelebA-64 are trained 250k iterations while models of ImageNet-256 trained 500k iters.

Method	Backbone	Params	ImageNet-256(conditional)		CIFAR-10(conditional)		CIFAR-10		CelebA-64	
			FID↓	IS↑	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑
SiT (origin) [30]	SiT-B/4	131M	56.69 / 28.22	24.67 / 55.65	12.55 / 8.60	8.22 / 8.90	17.41	7.82	7.04	2.76
			+0.0%/+0.0%	+0.0%/+0.0%	+0.0%/+0.0%	+0.0%/+0.0%	+0.0%	+0.0%	+0.0%	+0.0%
IP [31]	SiT-B/4	131M	54.48 / 25.98	26.05 / 60.20	12.90 / 8.30	8.18 / 8.97	15.60	7.96	5.23	2.72
SDSS [37]	SiT-B/4	131M	56.41 / 27.03	25.81 / 58.69	12.18 / 7.82	8.21 / 8.91	15.92	7.70	6.12	2.71
MDSS(4steps) [37]	SiT-B/4	131M	58.63 / 28.44	24.72 / 56.89	12.73 / 8.68	8.25 / 9.06	16.10	7.91	6.10	2.91
ReflexFlow w/o FC	SiT-B/4	131M	54.89 / 26.36	25.81 / 59.25	11.54 / 8.23	8.30 / 9.05	15.92	7.83	6.83	2.67
ReflexFlow w/o ADR	SiT-B/4	131M	52.30 / 25.39	26.55 / **60.43**	11.46 / 7.03	8.30 / **9.10**	15.99	**8.07**	4.96	**2.94**
ReflexFlow(ours)	SiT-B/4	131M	**51.58 / 25.07**	**26.62** / 58.94	**11.35 / 7.01**	**8.31** / 9.06	**14.82**	7.79	**4.53**	2.92
			-9.01%/ -11.16%	+7.90%/ +5.91%	-9.56%/ -18.49%	+1.09% /+1.80%	-14.88%	-0.38%	-35.65%	+5.80%

Algorithm 1 ReflexFlow: Training

Input: f is the model to predict velocity with parameters $\boldsymbol{\theta}$, dataset $\mathcal{D}$, learning rate η, ADR loss weight β_1, FC loss weight β_2, normalization coefficient α.
While $\boldsymbol{\theta}$ not converged **do**
  $\boldsymbol{\epsilon}\sim\mathcal{N}(0,I),\ \mathbf{x}_*\sim\mathcal{D},\ t_0,t_1\in[0,1],$ and $t_0>t_1$.
  $\mathbf{x}_{t_0}\leftarrow(1-t_0)\mathbf{x}_*+t_0\boldsymbol{\epsilon}, \mathbf{x}_{t_1}\leftarrow(1-t_1)\mathbf{x}_*+t_1\boldsymbol{\epsilon}$
  $\mathbf{v}_{t_0}\leftarrow f_{\boldsymbol{\theta}}(\mathbf{x}_{t_0},t_0), \mathbf{v}_{t_1}\leftarrow f_{\boldsymbol{\theta}}(\mathbf{x}_{t_1},t_1)$
  $\hat{\mathbf{x}}_{t_1}\leftarrow\mathbf{x}_{t_0}+(t_1-t_0)\mathbf{v}_{t_0}$ // Inferring one step
  $\hat{\mathbf{v}}_{t_1}\leftarrow f_{\boldsymbol{\theta}}(\hat{\mathbf{x}}_{t_1},t_1)$
  $\mathbf{v}_{\text{ADR}}\leftarrow\hat{\mathbf{x}}_{t_1}-\mathbf{x}_*$ // Anti-drift target
  $\mathcal{L}_{\text{ADR}}\leftarrow\mathbb{E}\left[\left\|\frac{\hat{\mathbf{v}}_{t_1}}{\|\hat{\mathbf{v}}_{t_1}\|_2}-\frac{\mathbf{v}_{\text{ADR},t}}{\|\mathbf{v}_{\text{ADR},t}\|_2}\right\|^2\right]$
  $\boldsymbol{\delta}\leftarrow\hat{\mathbf{v}}_{t_1}-\mathbf{v}_{t_1}$
  $\mathbf{W}_{\text{exp},t}^{(i,j)}\leftarrow 1+\alpha\frac{\boldsymbol{\delta}_{\text{exp},t}^{(i,j)}}{\sum_{i,j}\boldsymbol{\delta}_{\text{exp},t}^{(i,j)}}$, // Normalized as weight
  $\mathcal{L}_{\text{FC}}\leftarrow\mathbb{E}\left[\left\|\mathbf{W}_{\text{exp},t}\cdot\left(\mathbf{v}_\theta(\mathbf{x}_t,t)-(\boldsymbol{\epsilon}-\mathbf{x}_*)\right)\right\|^2\right]$
  $\mathcal{L}\leftarrow\beta_1\mathcal{L}_{\text{ADR}}+\beta_2\mathcal{L}_{\text{FC}}$
  $\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}-\eta\nabla_{\boldsymbol{\theta}}\mathcal{L}$

CIFAR-10 [21], as well as unconditional tasks on CIFAR-10 and CelebA-64 [28]. Following prior work [14, 31], unless otherwise specified, we generate 50K samples using 50 number of function evaluations (NFE, i.e., sampling steps). CIFAR-10 and CelebA-64 are processed in pixel space, while ImageNet-256 is encoded in the latent space of a pre-trained VAE tokenizer [38]. All models are trained from scratch on Ascend 910B NPUs. Based on grid search results, we set $\alpha=1$, $\beta_1=10$ and $\beta_2=1$ in all experiments. During training, all variants of our methods randomly select a pair of timesteps (t_0,t_1) with $t_0>t_1$ in each iteration. For ablation studies, we adopt SiT-B/4 as the default backbone unless stated otherwise.

Evaluation Metrics. We report FID [17] as the primary metric and follow standard practice by using the full training sets of CIFAR-10 and CelebA-64, and the testing set of ImageNet-256 as references. All values are computed using *torchfidelity* [33] to ensure fair comparison. In addition, Inception Score (IS) [40] is reported as a secondary metric. More details are provided in supplementary materials.

Frequency Analysis Details. In Eq. 16, the cutoff frequency is set to $\min(H,W)//8$, while the thresholds used in Eq. 18 are 20% and 25% respectively. This configuration yields the most distinct separation between low- and high-frequency regions, as illustrated in Fig. 5. All statistical results shown here are obtained from 5k randomly selected ImageNet-256 samples using SiT-B/4, and similar patterns are observed for other model variants.

Baselines. We compare our method with four baselines: SiT [30], IP [31], SDSS [37], and MDSS [37]. For IP, SDSS, and MDSS, which were originally developed under the DDPM framework, we adapt their core ideas to the flow matching setting. Specifically, IP perturbs the input with additional noise to simulate inference, SDSS aligns predictions with the target during single-step inference in training, and MDSS performs multi-step inference (we use 4 steps as reported to achieve the best performance [37]).

5.2 Quantitative Experiments

As the training environment differs, we reproduce the key experiments in Tab. 1 and summarize four main findings: *(i)* ReflexFlow consistently improves SiT, reducing FID by **9.01%/11.16%** on ImageNet-256 (w/o and w cfg=1.5), **9.56%/18.49%** on CIFAR-10, and **14.88%/35.65%** for unconditional CIFAR-10 and CelebA-64. This confirms its strong enhancement of FM generation quality. *(ii)* ADR surpasses MDSS and SDSS. Unlike SDSS [37], which uses ground-truth targets, ADR learns a reconstructed anti-drift target, demonstrating the benefit of dynamic target rectification. *(iii)* Both ADR and FC individually improve generation, and their combination in ReflexFlow achieves the best performance. *(iv)* Prior methods mitigate exposure bias via input perturbation (IP) or aligning original target after inference (SDSS, MDSS). ReflexFlow directly corrects prediction drift and compensates missing frequency components

at the source, achieving more accurate generation and significant gains over prior methods.

5.3 Qualitative Experiments

All methods use the same random seed and 50-NFE inference for fair comparison. As shown in Fig. 6, baseline methods exhibit clear visual artifacts: the second and fourth rows show blurred object–background boundaries, while the first and third rows display structural distortions and missing details. These issues arise from exposure bias accumulating during HFT, causing confusion between the primary object and surrounding context. In contrast, ReflexFlow generates images with more coherent structures and consistent textures by suppressing prediction drift and compensating missing frequency components, demonstrating its effectiveness in mitigating exposure bias.

6 Ablation Study

This section presents ablation studies on various components of our proposed method and details the implementation hyperparameters.

6.1 Effectiveness of Normalization for Anti-Drift Target

Table 2. **Learning direction is stabler.** Results are reported using a SiT-B/4 model trained for 250k iterations on the CIFAR-10 conditional generation task under 50-NFE without cfg.

Method	ADR Normalization	Direction	CIFAR-10(cond)	
			FID↓	IS↑
SiT	/	/	12.55	8.22
ReflexFlow	/	Anti-Drift	317.31	1.38
ReflexFlow	Rescale	Anti-Drift	20.20	7.93
ReflexFlow	Unite Length	Directly Straight	<u>12.15</u>	<u>8.29</u>
ReflexFlow (ours)	Unite Length	Anti-Drift	**11.35**	**8.31**

In this section, we analyze why normalizing the anti-drift target is essential and demonstrate its importance through empirical evidence.

Without normalization, the anti-drift target misguides the model toward noisy outputs. Removing both the flow-matching loss and the normalization, we make the model to directly predict the anti-drift target:

$$\begin{aligned} \mathbf{v}_{\text{ADR}} &= \hat{\mathbf{x}}_{t_1} - \mathbf{x}_*, \\ \text{where} \quad \hat{\mathbf{x}}_{t_1} &= \mathbf{x}_{t_0} + (t_1 - t_0)\mathbf{v}_{t_0}. \end{aligned} \tag{23}$$

While this target may appear to mitigate exposure bias, it fundamentally fails to fully remove noise. Assuming the model makes perfect prediction, the endpoint of the inference trajectory becomes:

$$\begin{aligned} \hat{\mathbf{x}}_* &= \boldsymbol{\epsilon} + \int_1^0 (\hat{\mathbf{x}}_t - \mathbf{x}_*)\, dt \\ &= \mathbf{x}_* + \boldsymbol{\epsilon} - \int_0^1 \hat{\mathbf{x}}_t\, dt. \end{aligned} \tag{24}$$

In practice, the integral term $\int_0^1 \hat{\mathbf{x}}_t\, dt$ can not cancel the noise $\boldsymbol{\epsilon}$, because $\hat{\mathbf{x}}_t$ is far from Gaussian distribution. As shown in Fig. 2 (left), the loss under biased inputs grows along the denoising trajectory. This loss is expected to stay approximately constant only if $\hat{\mathbf{x}}_t$ follows a Gaussian distribution. Consequently, even with a perfectly learned target, the model still lacks the capability to recover the ground truth sample $\mathbf{x}_*$, which explains the poor performance of directly learning Eq. (23) reported in the second row of Tab. 2.

Rescaling the anti-drift target introduces training instability. Directly learning the anti-drift target is problematic, as $\hat{\mathbf{x}}_{t_1}$ deviates substantially from a Gaussian distribution at different times t_1. A natural idea is to rescale the anti-drift target so that its magnitude matches the original objective $(\boldsymbol{\epsilon} - \mathbf{x}_*)$, which also reduces the variation in the distribution of $\hat{\mathbf{x}}_{t_1}$ induced by different timesteps. Concretely, we divide by the effective time length:

$$\mathbf{v}_{\text{Rescaled ADR}} = \frac{\hat{\mathbf{x}}_{t_1} - \mathbf{x}_*}{t_1}. \tag{25}$$

This target velocity alignment ensures that integrating the rescaled direction from 1 to 0 achieves a displacement of the correct magnitude. Conceptually, we can view this as estimating a predicted biased noise $\boldsymbol{\epsilon}'$:

$$\boldsymbol{\epsilon}' = \frac{\hat{\mathbf{x}}_{t_1} - \mathbf{x}_*}{t_1} + \mathbf{x}_*, \tag{26}$$

and integrating $\boldsymbol{\epsilon}' - \mathbf{x}_*$ during inference to reduce timestep-dependent variation.

However, training samples t_1 over the full interval $[0, 1]$. As $t_1 \to 0$, the factor $1/t_1$ in Eq. (25) diverges, causing exploding losses, numerical instability, and NaN gradients (even if NaNs are later filtered). This instability explains the poor results in the third row of Tab. 2.

Normalization ensures directional guidance for anti-drift rectification. To avoid unstable scaling while keeping the directional information, we normalize both the predicted velocity and the anti-drift target:

$$\hat{\mathbf{v}}_{t_1} = \frac{\hat{\mathbf{v}}_{t_1}}{\|\hat{\mathbf{v}}_{t_1}\|_2}, \quad \mathbf{v}_{\text{ADR}} = \frac{\hat{\mathbf{x}}_{t_1} - \mathbf{x}_*}{\|\hat{\mathbf{x}}_{t_1} - \mathbf{x}_*\|_2}. \tag{27}$$

We then add a directional regularization term to the flow-matching objective:

$$\mathcal{L}_{\text{ADR}} = \mathbb{E}\left[\left\| \frac{\hat{\mathbf{v}}_{t_1}}{\|\hat{\mathbf{v}}_{t_1}\|_2} - \frac{\hat{\mathbf{x}}_{t_1} - \mathbf{x}_*}{\|\hat{\mathbf{x}}_{t_1} - \mathbf{x}_*\|_2} \right\|_2^2\right]. \tag{28}$$

This preserves the flow-matching framework while explicitly guiding the predicted direction toward the true endpoint $\mathbf{x}_*$. Importantly, although Eq. (28) enforces only directional consistency, the inference velocity retains the correct magnitude scale inherited from the flow-matching term. Consequently, integrating the velocity from 1 to 0 achieves meaningful and stable results as illustrated of the best performance of the Tab. 2.

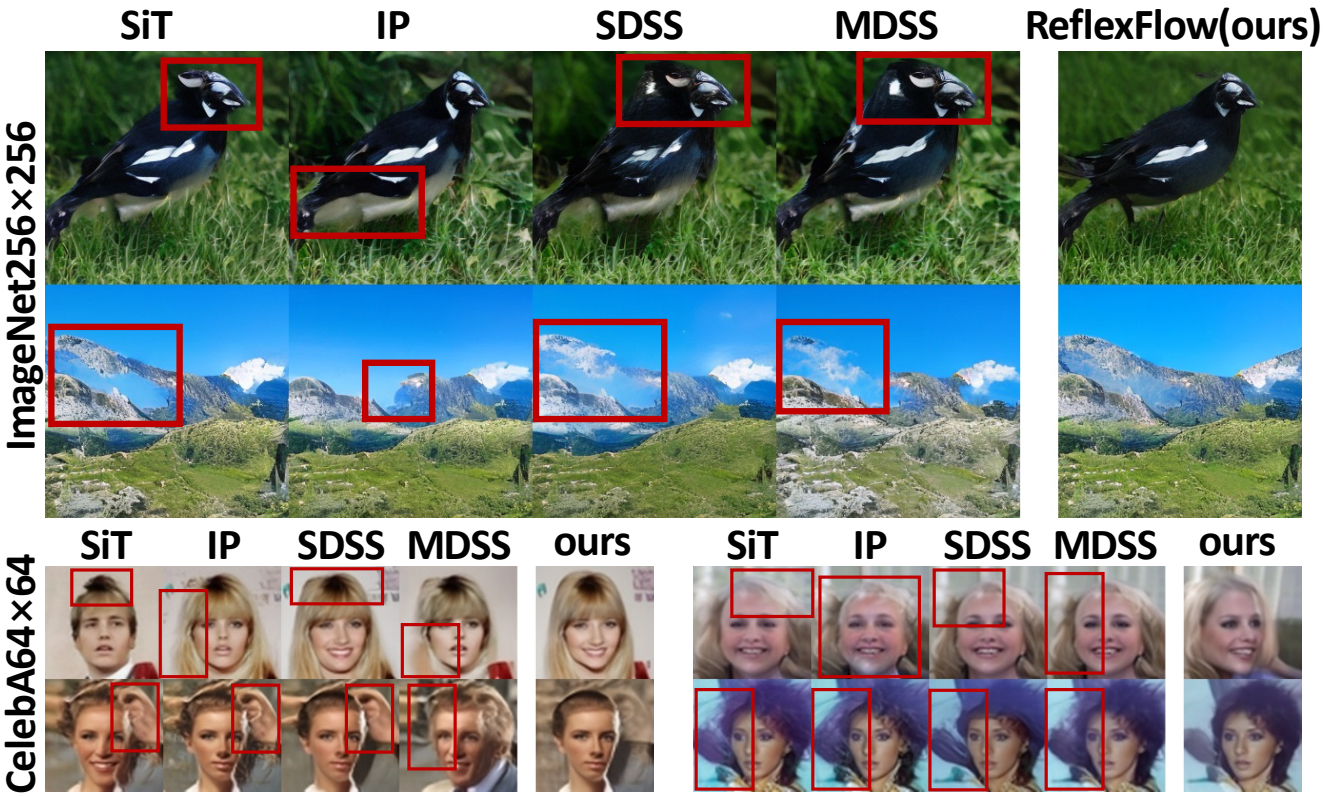

Figure 6. **Qualitative Comparison.** ReflexFlow produces most realistic samples on ImageNet-256 and CelebA-64, clearly outperforming baselines, which exhibit blurriness or distorted details, as highlighted with red boxes.

Table 3. **Effectiveness of exposure bias for frequency compensation.** We compare different weighting strategies for frequency compensation, including low-pass, high-pass, original-image, original-noise and exposure bias weights. Results are reported using a SiT-B/4 model trained for 250k iterations on the CIFAR-10 conditional generation task without cfg.

Method	Weight Type	Objective	CIFAR-10(cond)	
			FID↓	IS↑
SiT [30]	/	$\mathcal{L}_{FM}$	12.55	8.22
ReflexFlow w/o FC + High Pass	High Pass	on $\mathcal{L}_{FM}$	13.14	8.06
ReflexFlow w/o FC + Low Pass	Low Pass	on $\mathcal{L}_{FM}$	<u>11.94</u>	<u>8.29</u>
ReflexFlow w/o FC + Original Image	$\mathbf{x}_*$	on $\mathcal{L}_{FM}$	12.49	8.20
ReflexFlow w/o FC + Original Noise	$\boldsymbol{\epsilon}$	on $\mathcal{L}_{FM}$	13.66	8.04
ReflexFlow(ours)	Exposure Bias	on $\mathcal{L}_{FM}$	**11.35**	**8.31**

We further investigate whether the model can directly learn a target direction that returns to the ideal path at timestep t_1 straightly. Specifically, we train on the direction $\hat{\mathbf{x}}_t - \mathbf{x}_{t_1}$ and apply unit-length normalization. The result is reported in the fourth row of Tab. 2. This experiment highlights the importance of unit-length normalization and also reveals that directly learning a downward direction can conflict with the flow-matching objective. The anti-drift direction remains compatible: under perfect prediction $\mathbf{x}_{t_1}$ would already lie on the ideal path, causing the anti-drift vector to degenerate back to the original flow matching direction $\boldsymbol{\epsilon} - \mathbf{x}_*$. However, the directly downward target collapses to a zero vector $\mathbf{0}$ in this situation, contradicting the flow matching objective and leading to suboptimal performance.

6.2 Effectiveness of Exposure Bias for Frequency Compensation

Fig. 2 (right) shows that the frequency deficiency of the model varies across timesteps: it lacks low-frequency components during HFT, but accumulates excessive low-frequency content during LFT. As shown in Tab. 3, methods that directly inject fixed low- or high-frequency components of the original image (rows two and three), or inject the original image or noise directly (rows four and five), only correct part of the frequency deficiency while amplifying the opposite deficiency of the model. In contrast, exposure bias provides a dynamic signal that reflects the instantaneous frequency shortfall of the model. It adds low-frequency information when the model lacks it in HFT, and naturally reduces such supplementation when low-frequency components become dominant in LFT as shown in Fig. 3 (left). This adaptive behavior leads to the best overall performance in Tab. 3.

It is also worth noting that injecting only low-frequency information consistently outperforms injecting only high-frequency information (e.g., low-pass vs. high-pass, orig-

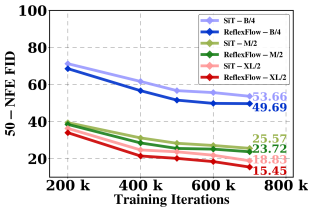

Figure 7. **ReflexFlow exhibits promising scalability with respect to model size.** All models are trained from scratch and evaluated without cfg on ImageNet-256.

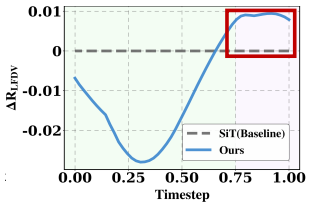

Figure 8. **Ours vs. baseline: compensates the missing low-frequency components during HFT (marked by the red box).** Frequency analysis after trained for 500k iterations.

inal image vs. original noise). This indicates that low-frequency learning has a larger impact on final performance. Since exposure bias dynamically compensates the low-frequency content the model lacks, these results further highlight the effectiveness of exposure bias as a frequency-aware corrective signal.

Table 4. **Hyperparameter ablation under 50-NFE.** Results are reported using a SiT-B/4 model trained for 500k iterations on ImageNet-256. Conditional generation without cfg.

Method	Coefficient	Objective	Normalization	ImageNet256	
				FID↓	IS↑
ADR	$\beta_1 = 1$	Regularizer	p=2	55.75	25.57
ADR	$\beta_1 = 10$	Regularizer	p=1	56.34	25.02
ADR	$\beta_1 = 10$	Regularizer	p=2	**54.89**	**25.81**
ADR	$\beta_1 = 50$	Regularizer	p=2	55.92	25.33
FC	$\alpha = 0.5$	on weight	/	54.58	26.47
FC	$\alpha = 1.0$	on weight	/	**52.30**	**26.55**
FC	$\alpha = 5.0$	on weight	/	53.35	26.48
FC	$\beta_2 = 0.5$	on $\mathcal{L}_{FM}$	/	53.61	26.50
FC	$\beta_2 = 1.0$	on $\mathcal{L}_{FM}$	/	**52.30**	**26.55**
FC	$\beta_2 = 5.0$	on $\mathcal{L}_{FM}$	/	54.62	26.46

6.3 Hyperparameter Ablation of Our Methods

The results of Tab. 4 shows using L_2 normalization (Euclidean norm, $p = 2$) would be more stable than L_1 normalization ($p = 1$), and $\beta_1 = 10$ produces optimal performance on ADR, L_2 normalization and $\beta_1 = 10$ are thus adopted for all experiments.

For FC hyperparameters, the weight of the exposure bias in the original flow matching loss is determined by normalizing the squared L_2 norm of the exposure bias. Ablation studies show that $\alpha = 1$ achieves the best performance. We further conduct ablations on β_2 and find that $\beta_2 = 1$ also performs best. Therefore, these values are used consistently throughout the paper.

7 Discussion

In this section, we analyze the effectiveness of our method in mitigating exposure bias based on experimental results.

Our method bridges frequency gaps. As shown in Fig. 8,

Table 5. **Scalability of ReflexFlow on ImageNet-256.** 50-NFE generation without/with cfg=1.5. Models are trained from scratch.

Method	Backbone	Params	Iters	ImageNet-256	
				FID↓	IS↑
SiT [30]	SiT-B/4	131M	500k	56.69 / 28.22	24.67 / 55.65
ReflexFlow(ours)	SiT-B/4	131M	500k	**51.58 / 25.07**	**26.62 / 58.94**
				- 9.01% / - 11.16%	+ 7.90% / + 5.91%
SiT [30]	SiT-M/2	308M	500k	28.29 / 7.87	49.22 / 129.82
ReflexFlow(ours)	SiT-M/2	308M	500k	**25.50 / 7.12**	**51.64 / 132.87**
				- 9.86% / - 9.53%	+ 4.92% / + 2.35%
SiT [30]	SiT-XL/2	675M	500k	23.67 / 5.87	57.57 / 154.05
ReflexFlow(ours)	SiT-XL/2	675M	500k	**20.14 / 5.20**	**64.11 / 166.00**
				- 14.91% / - 11.41%	+ 11.36% / + 7.76%
SiT [30]	SiT-XL/2	675M	700k	18.83 / 4.36	68.93 / 183.58
ReflexFlow(ours)	SiT-XL/2	675M	700k	**15.45 / 4.16**	**77.80 / 199.59**
				- 17.95% / - 4.59%	+ 12.87% / + 8.72%

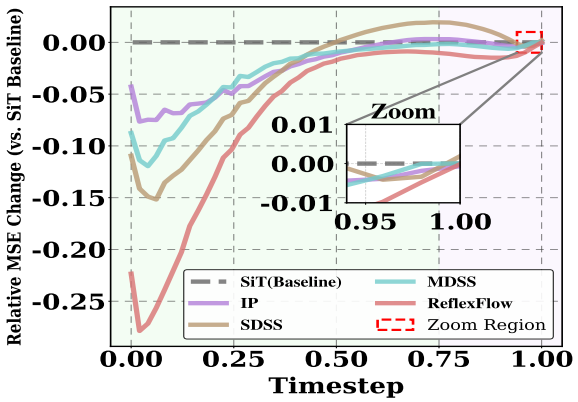

Figure 9. **Quantitative analysis of exposure bias over timesteps for models trained with SiT-B/4 for 500k iterations.** We measure the MSE between predicted and ground-truth velocities for all methods, then report the difference relative to the SiT baseline. ReflexFlow consistently achieves the largest reduction, highlighting its superior robustness.

Table 6. **The effect of multi-NFE sampling on ImageNet-256.** Entries are reported without cfg. Models are trained from scratch.

Steps	SiT		IP		SDSS		MDSS		ReflexFlow	
	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑
30	58.57	24.70	54.18	**26.21**	58.46	25.80	60.78	24.78	**52.37**	25.84
50	56.69	24.67	54.48	26.05	56.41	25.81	58.63	24.72	**51.58**	**26.62**
100	55.58	24.48	53.38	26.01	55.17	25.78	57.18	24.64	**51.15**	**26.51**
250	55.08	24.30	52.93	25.83	54.54	25.64	56.48	24.48	**50.97**	**26.33**
500	54.96	24.24	52.79	25.76	54.39	25.56	56.27	24.41	**50.92**	**26.30**

we compare the R_{LFDV} of predicted velocities between our method and the original SiT baseline. Our method provides more low-frequency components at HFT, alleviating the low-frequency deficiency.

ReflexFlow outperforms prior methods in mitigating exposure bias. We quantify exposure bias by computing the relative MSE changes (with vanilla SiT as the baseline) between predicted and ground-truth velocities during inference (Fig. 9). Compared to all other methods, ReflexFlow achieves the lowest bias across all the timesteps, demonstrating its superior capability to mitigate exposure bias.

Scalability study of ReflexFlow models. Tab. 5 reports

FID and IS results, and Fig. 7 shows the training durations of ReflexFlow across different model scales. ReflexFlow consistently outperforms the original SiT models and exhibits strong scalability in generative quality. Given the consistent emergence of scaling laws in AIGC, ReflexFlow is well-positioned to leverage these trends, enabling scalable, high-fidelity generation across domains.

Comparison under varying inference steps. Potential prediction bias can accumulate over time during inference. Increasing inference steps often amplifies exposure bias and degrades generation quality. To assess this effect, we evaluate all methods under various step settings (30, 50, 100, 250, 500). ReflexFlow consistently achieves the lowest FID and highest IS across nearly all settings, maintaining stable performance as the sampling step increases, while baselines degrade noticeably in Tab. 6. These results highlight the robustness of ReflexFlow in mitigating exposure bias under extended inference. While baselines steadily drops on IS, ReflexFlow remains strong up to 50 steps and even at 500 steps outperforms the best steps of all baselines.

8 Conclusion

In this work, we present the first systematic study of exposure bias in Flow Matching, revealing its root causes from both prediction drift and frequency deficiency perspectives. Based on these insights, we propose ReflexFlow, a unified framework comprising Anti-Drift Rectification (ADR) to correct prediction drift and Frequency Compensation (FC) to adaptively address missing frequency components via self-regulated negative feedback. Our approach is simple, model-agnostic, and broadly compatible with all flow matching frameworks. Extensive experiments across multiple datasets demonstrate that ReflexFlow more effectively mitigates exposure bias and consistently improves generation quality over baselines.

ReflexFlow: Rethinking Learning Objective for Exposure Bias Alleviation in Flow Matching

Supplementary Material

Contents of Appendix

A Notations **13**

B Proofs of Propositions **14**
B.1. Error Bounded by FM Target 14
B.2. Error Bounded by ADR Target 15
B.3. Relating FM and ADR Residuals 16

C More Experiments **17**
C.1. Multi-Step Scheduled Sampling Training of ADR 17
C.2. Extended-Training Stability and Effectiveness of ReflexFlow 18
C.3. Multi-NFE on ImageNet-256 with CFG . . . 18

D More Related Work **18**

E Additional Experimental Details **18**

F. Baseline Implementation Details **18**

G More Qualitative Comparison on ImageNet-256 and CelebA-64 **19**

H Generated Samples on ImageNet-256 under 50-NFE. **20**

I. More Frequency-Aware Visualization on ImageNet-256 **20**

A. Notations

Symbol	Description
$\mathbf{x}_*$	Target data sample
$\mathbf{x}_t$	The ideal forward linear interpolation between data and Gaussian noise at timestp t
$\hat{\mathbf{x}}_t$	The reverse predicted biased sample at timestep t that contained potential drift
$\hat{\mathbf{x}}_*$	The final predicted biased endpoint
$\boldsymbol{\epsilon}$	Gaussian noise
$\mathbf{v}_{\text{target}}$	Original learning target $\boldsymbol{\epsilon}$ - $\mathbf{x}_*$
$\mathbf{v}_t$	The predicted velocity from the model when input the forward perturbed $\mathbf{x}_t$
$\hat{\mathbf{v}}_t$	The predicted velocity from the model when input the reverse predicted biased input $\hat{\mathbf{x}}_{t_k}$
$\mathbf{v}_{\text{ADR},t}$	The reconstructed anti-drift rectification target
$\mathbf{V}$	The frequency-domain representation of velocity map
$\mathbf{v}_{\text{low/high}}$	The low / high frequency components on velocity map after IFFT transforming
$\mathbf{M}_{\text{LFR/HFR}}$	The dominant low / high frequency mask of $\mathbf{v}_{\text{target}}$
t	The timestep and $t \in [0, 1]$
$t_{0/1}$	The start timestep / inferred timestep of inference in one step training-time inference process, and $t_0 > t_1$
$\boldsymbol{\theta}$	Model parameters
$\mathbf{v}_{\boldsymbol{\theta}}(\cdot)$	The function with parameter $\boldsymbol{\theta}$ for predicting velocity
$p(\cdot)$	The distribution function
$\mathcal{L}$	The overall loss
$\mathcal{L}_{\text{ADR}}$	The anti-drift rectification loss
$\mathcal{L}_{\text{FC}}$	The frequency compensation loss
$\boldsymbol{\mathcal{L}}$	The loss map with $H \times W$
$\tilde{\boldsymbol{\mathcal{L}}}$	The normalization result of $\boldsymbol{\mathcal{L}}$
$\mathcal{N}$	Normal distribution
$\boldsymbol{\delta}_{\text{exp},t}$	The exposure bias at timestep t
h	Discrete timestep in theoretical analysis
τ_k	The timestep setting for theoretical analysis starting from 0 to 1 as the denoising path at time k
K	The final sampling timestep
$\mathbf{e}_t$	The potential drift at timestep t between $\hat{\mathbf{x}}_{t_k}$ and $\mathbf{x}_{t_k}$
$\mathbf{r}_t$	The residual between ground truth velocity and predicted biased velocity
$\boldsymbol{\phi}_\tau$	The ADR residual between predicted biased velocity and anti-drift velocity
R_{LFDV}	The low-frequency components dominant ratio on velocity
R_{LFDL}	The ratio in the loss that emphasizes the low-frequency dominant components of $\mathbf{v}_{\text{target}}$
DFT / FFT / IFFT	Discrete Fourier Transform / Fast Fourier Transform / Inverse Fast Fourier Transform
$\mathbf{F}_{\text{LP/HP}}$	The low-pass / high-pass filter
HFT / LFT	High-frequency timesteps / low-frequency timesteps
$\mathbf{W}_{\text{exp},t}$	The frequency-aware weight modulated by exposure bias
(u, v)	The coordinate of the frequency-domain signal
R	The real part of DFT result
I	The imaginary part of DFT result
$\mathcal{D}$	Training dataset
$H \times W$	The height and width shape of a sample
α	The coefficient modulating the influence of exposure bias on the frequency-aware weight
β_1	The hyperparameter of ADR loss
β_2	The hyperparameter of FC loss
η	Learning rate
$\boldsymbol{\epsilon}'$	The predicted Gaussian noise

B. Proofs of Propositions

B.1. Error Bounded by FM Target

Proposition 4.1. *The bound for the expected final sampling error $\mathbb{E}\|\mathbf{e}_{\tau_K}\|$ under the Flow Matching (FM) objective can be expressed as the sum of the integrated ground truth velocity residual and the integrated exposure bias, which is formulated as follows:*

$$\mathbb{E}\|\mathbf{e}_{\tau_K}\| \leq \underbrace{\int_0^1 \sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2}d\tau}_{\text{Integrated GT Velocity Residual}} + \underbrace{\int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau_k}\|^2}d\tau}_{\text{Integrated Exposure Bias}}. \tag{29}$$

where $\mathbf{e}_{\tau_k} := \hat{\mathbf{x}}_{\tau_k} - \mathbf{x}_{\tau_k}$ is the potential error at k-th timestep between the reverse predicted biased sample $\hat{\mathbf{x}}_{\tau_k}$ and the forward ideal interpolated sample $\mathbf{x}_{\tau_k}$, and the final timestep is K at the predicted endpoint $\hat{\mathbf{x}}_*$. We define the ground truth velocity residual as $\mathbf{r}_{\tau_k} := \mathbf{v}_\theta(\mathbf{x}_{\tau_k}, \tau_k) - (\boldsymbol{\epsilon} - \mathbf{x}_*)$, which measures the prediction drift of model from the ground-truth target, and the exposure bias $\boldsymbol{\delta}_{exp,\tau_k}$ is defined in Eq. 6.

Proof of Proposition 4.1. We consider a discrete time formulation with step size $h = \frac{1}{K}$ and nodes $\tau_k = kh$ for $k = 0, \ldots, K$. The denoising process starts at $\tau = 0$ with coupled initial states $\hat{\mathbf{x}}_0 = \mathbf{x}_0$, and the initial deviation is $\mathbf{e}_0 = \mathbf{0}$. After applying a Euler integrator to the predicted velocity and ground truth velocity, the discrete updates read:

$$\begin{aligned} \hat{\mathbf{x}}_{\tau_{k+1}} &= \hat{\mathbf{x}}_{\tau_k} + h\,\mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k}, \tau_k), \\ \mathbf{x}_{\tau_{k+1}} &= \mathbf{x}_{\tau_k} + h\,\mathbf{v}_{\text{target}}(\mathbf{x}_{\tau_k}, \tau_k). \end{aligned} \tag{30}$$

Here $\mathbf{v}_\theta$ denotes the learned velocity and $\mathbf{v}_{\text{target}}$ denotes the ground truth target velocity along the ideal interpolation path. We then review the definition of exposure bias $\boldsymbol{\delta}_{\text{exp},\tau_k}$:

$$\boldsymbol{\delta}_{\text{exp},\tau_k} = \mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k}, \tau_k) - \mathbf{v}_\theta(\mathbf{x}_{\tau_k}, \tau_k). \tag{31}$$

As we take the typical schedule of Flow Matching, $\mathbf{v}_{\text{target}}(\mathbf{x}_{\tau_k}, \tau_k) = \boldsymbol{\epsilon} - \mathbf{x}_*$ is constant along the path. We then define the ground truth velocity residual $\mathbf{r}_{\tau_k}$ as:

$$\mathbf{r}_{\tau_k} := \mathbf{v}_\theta(\mathbf{x}_{\tau_k}, \tau_k) - \mathbf{v}_{\text{target}}(\mathbf{x}_{\tau_k}, \tau_k). \tag{32}$$

After establishing the identities above, we proceed to express the evolution of the sampling error. Substituting the update rules of $\hat{\mathbf{x}}_{\tau_{k+1}}$ (Eq. 30) and $\mathbf{x}_{\tau_{k+1}}$ and the definition of $\mathbf{e}_{\tau_k} := \hat{\mathbf{x}}_{\tau_k} - \mathbf{x}_{\tau_k}$, and using $\tau_k = kh$ and $h = \frac{1}{K}$, we can rewrite the error at the next step $\mathbf{e}_{\tau_{k+1}}$ as follows:

$$\begin{aligned} \mathbf{e}_{\tau_{k+1}} &= \hat{\mathbf{x}}_{\tau_{k+1}} - \mathbf{x}_{\tau_{k+1}} \\ &= \big(\hat{\mathbf{x}}_{\tau_k} + h\,\mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k}, \tau_k)\big) - \big(\mathbf{x}_{\tau_k} + h\,\mathbf{v}_{\text{target}}(\mathbf{x}_{\tau_k}, \tau_k)\big) \\ &= (\hat{\mathbf{x}}_{\tau_k} - \mathbf{x}_{\tau_k}) + h[\mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k}, \tau_k) - \mathbf{v}_\theta(\mathbf{x}_{\tau_k}, \tau_k)] \\ &\quad + h[\mathbf{v}_\theta(\mathbf{x}_{\tau_k}, \tau_k) - \mathbf{v}_{\text{target}}(\mathbf{x}_{\tau_k}, \tau_k)] \\ &= \mathbf{e}_{\tau_k} + h\,\underbrace{(\mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k}, \tau_k) - \mathbf{v}_\theta(\mathbf{x}_{\tau_k}, \tau_k))}_{\boldsymbol{\delta}_{\text{exp},\tau_k}} \\ &\quad + h\,\underbrace{(\mathbf{v}_\theta(\mathbf{x}_{\tau_k}, \tau_k) - \mathbf{v}_{\text{target}}(\mathbf{x}_{\tau_k}, \tau_k))}_{\mathbf{r}_{\tau_k}} \\ &= \mathbf{e}_{\tau_k} + h\,\boldsymbol{\delta}_{\text{exp},\tau_k} + h\,\mathbf{r}_{\tau_k}. \end{aligned} \tag{33}$$

This completes the step-wise recursion. Setting $\mathbf{e}_0 = 0$ and summing (33) over all steps yields:

$$\mathbf{e}_{\tau_K} = h\sum_{k=0}^{K-1}\mathbf{r}_{\tau_k} + h\sum_{k=0}^{K-1}\boldsymbol{\delta}_{\text{exp},\tau_k}. \tag{34}$$

Taking norms on both sides of the recursion and invoking the triangle inequality for vector norms, we obtain:

$$\|\mathbf{e}_{\tau_K}\| \leq h\sum_{k=0}^{K-1}\|\mathbf{r}_{\tau_k}\| + h\sum_{k=0}^{K-1}\|\boldsymbol{\delta}_{\text{exp},\tau_k}\|, \tag{35}$$

where $\|\cdot\|$ denotes the Euclidean norm, and the inequality follows from the subadditivity property $\|a+b\| \leq \|a\| + \|b\|$.

We then take expectation and apply the Cauchy–Schwarz inequality in the form $\mathbb{E}\|Z\| \leq \sqrt{\mathbb{E}\|Z\|^2}$. We obtain:

$$\begin{aligned} \mathbb{E}\|\mathbf{e}_{\tau_K}\| &\leq h\sum_{k=0}^{K-1}\mathbb{E}\|\mathbf{r}_{\tau_k}\| + h\sum_{k=0}^{K-1}\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau_k}\| \\ &\leq h\sum_{k=0}^{K-1}\sqrt{\mathbb{E}\|\mathbf{r}_{\tau_k}\|^2} + h\sum_{k=0}^{K-1}\sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau_k}\|^2}. \end{aligned} \tag{36}$$

Since FM is defined on a continuous-time path, we move from the discrete sum to its continuous counterpart $h\sum_{k=0}^{K-1} \to \int_0^1 d\tau$. This gives:

$$\mathbb{E}\|\mathbf{e}_{\tau_K}\| \leq \int_0^1 \sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2}\,d\tau + \int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau}\|^2}\,d\tau. \tag{37}$$

Under vanilla FM training, the learning objective directly minimizes only the first integral, i.e., the ground truth velocity residual $\int_0^1 \sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2}\,d\tau$. However, the second integral $\int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau}\|^2}\,d\tau$, which captures exposure bias, remains uncontrolled during training. As a result, exposure bias accumulates across timesteps and contributes linearly to the global sampling error. This analysis highlights that conventional FM training inherently overlooks a dominant source of sampling error, underscoring the need to mitigate exposure bias. □

B.2. Error Bounded by ADR Target

Proposition 4.3. *The bound form with ADR of the expected final sampling error $\mathbb{E}\|\mathbf{e}_{\tau_K}\|$ is governed by the integrated ADR residual and a constant path-geometry term:*

$$\mathbb{E}\|\mathbf{e}_{\tau_K}\| \le \underbrace{\int_0^1 e^{(1-\tau)}\sqrt{\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2}d\tau}_{\text{Integrated ADR Residual}} + \underbrace{(e-2)\|\boldsymbol{\epsilon}-\mathbf{x}_*\|}_{\text{Path-Geometry Constant}}, \tag{38}$$

where $\boldsymbol{\phi}_\tau := \mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k},\tau) - (\hat{\mathbf{x}}_{\tau_k}-\mathbf{x}_*)$ is the ADR residual. *This term represents the prediction drift of velocity on the biased $\hat{\mathbf{x}}_{\tau_k}$ relative to the anti-drift target $(\hat{\mathbf{x}}_{\tau_k}-\mathbf{x}_*)$, which we introduce in Eq. 8.*

proof of Proposition 4.3. We begin by analyzing the discrepancy between the predicted velocity and the anti-drift target velocity. For each step, we define the ADR residual as:

$$\boldsymbol{\phi}_{\tau_k} := \mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k},\tau_k) - (\hat{\mathbf{x}}_{\tau_k}-\mathbf{x}_*). \tag{39}$$

This residual captures the prediction drift evaluated on the biased state $\hat{\mathbf{x}}_{\tau_k}$ relative to the anti-drift target.

We next observe that the ground truth residual $\mathbf{r}_{\tau_k}$ can be decomposed into two components: the exposure bias term $\boldsymbol{\delta}_{\text{exp},\tau_k}$ and the ADR residual $\boldsymbol{\phi}_{\tau_k}$. A direct decomposition yields:

$$\begin{aligned}\mathbf{r}_{\tau_k} &= \mathbf{v}_\theta(\mathbf{x}_{\tau_k},\tau_k) - (\boldsymbol{\epsilon}-\mathbf{x}_*)\\ &= \left[\mathbf{v}_\theta(\mathbf{x}_{\tau_k},\tau_k) - \mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k},\tau_k)\right]\\ &\quad + \left[\mathbf{v}_\theta(\hat{\mathbf{x}}_{\tau_k},\tau_k) - (\hat{\mathbf{x}}_{\tau_k}-\mathbf{x}_*)\right]\\ &\quad + \left[(\hat{\mathbf{x}}_{\tau_k}-\mathbf{x}_*) - (\boldsymbol{\epsilon}-\mathbf{x}_*)\right]\\ &= -\boldsymbol{\delta}_{\text{exp},\tau_k} + \boldsymbol{\phi}_{\tau_k} - (\boldsymbol{\epsilon}-\hat{\mathbf{x}}_{\tau_k}).\end{aligned} \tag{40}$$

Substituting Eq. 40 into Eq. 33 leads to a cancellation of the exposure bias term $\boldsymbol{\delta}_{\text{exp},\tau_k}$:

$$\begin{aligned}\mathbf{e}_{\tau_{k+1}} &= \mathbf{e}_{\tau_k} + h\,\boldsymbol{\delta}_{\text{exp},\tau_k}\\ &\quad + h\left[-\boldsymbol{\delta}_{\text{exp},\tau_k} + \boldsymbol{\phi}_{\tau_k} - (\boldsymbol{\epsilon}-\hat{\mathbf{x}}_{\tau_k})\right]\\ &= \mathbf{e}_{\tau_k} + h\,\boldsymbol{\phi}_{\tau_k} + h(\hat{\mathbf{x}}_{\tau_k}-\boldsymbol{\epsilon}).\end{aligned} \tag{41}$$

We rewrite the remaining term using:

$$\begin{aligned}\hat{\mathbf{x}}_{\tau_k} - \boldsymbol{\epsilon} &= \mathbf{e}_{\tau_k} + (\mathbf{x}_{\tau_k}-\boldsymbol{\epsilon}),\\ \mathbf{x}_{\tau_k} - \boldsymbol{\epsilon} &= \tau_k(\mathbf{x}_*-\boldsymbol{\epsilon}).\end{aligned} \tag{42}$$

And we obtain the recursion as:

$$\mathbf{e}_{\tau_{k+1}} = (1+h)\,\mathbf{e}_{\tau_k} + h\,\boldsymbol{\phi}_{\tau_k} + h\,\tau_k\,(\mathbf{x}_*-\boldsymbol{\epsilon}). \tag{43}$$

The recursion in Eq. 43 can be rewritten as a first-order linear difference equation:

$$\mathbf{e}_{\tau_{k+1}} - \mathbf{e}_{\tau_k} = h\,\mathbf{e}_{\tau_k} + h\,\boldsymbol{\phi}_{\tau_k} + h\,\tau_k(\mathbf{x}_*-\boldsymbol{\epsilon}). \tag{44}$$

Taking the limit $h \to 0$ with the initialization $\mathbf{e}_0 = \mathbf{0}$, the discrete recursion converges to the following first-order ordinary differential equation:

$$\frac{d}{d\tau}\mathbf{e}_\tau = \mathbf{e}_\tau + \boldsymbol{\phi}_\tau + \tau(\mathbf{x}_*-\boldsymbol{\epsilon}). \tag{45}$$

This ODE Eq. 45 admits a closed-form solution. Multiplying both sides by the integrating factor $e^{-\tau}$, where e denotes Euler's number, the base of the natural logarithm. This yields:

$$\frac{d}{d\tau}\left(e^{-\tau}\mathbf{e}_\tau\right) = e^{-\tau}\Big[\boldsymbol{\phi}_\tau + \tau(\mathbf{x}_*-\boldsymbol{\epsilon})\Big], \tag{46}$$

and integrating from 0 to τ gives,

$$\mathbf{e}_\tau = e^{\tau}\int_0^\tau e^{-s}\Big[\boldsymbol{\phi}_s + s(\mathbf{x}_*-\boldsymbol{\epsilon})\Big]\,ds. \tag{47}$$

Evaluating the solution of Eq. 47 at the end of the denoising path $\tau = 1$ yields:

$$\begin{aligned}\mathbf{e}_{\tau_K} = \mathbf{e}_1 &= \int_0^1 e^{1-s}\,\boldsymbol{\phi}_s\,ds + \left(\int_0^1 e^{1-s}\,s\,ds\right)(\mathbf{x}_*-\boldsymbol{\epsilon})\\ &= \int_0^1 e^{1-\tau}\,\boldsymbol{\phi}_\tau\,d\tau + \underbrace{\int_0^1 e^{1-\tau}\tau\,d\tau}_{=e-2}(\mathbf{x}_*-\boldsymbol{\epsilon}).\end{aligned} \tag{48}$$

We then bound the expected terminal error. Throughout, we use the Euclidean norm $\|\mathbf{z}\| := \sqrt{\langle\mathbf{z},\mathbf{z}\rangle}$ for any $\mathbf{z}\in\mathbb{R}^d$. Specifically, starting from the expression of $\mathbf{e}_1$ derived above, we first apply the triangle inequality for vectors, $\|\mathbf{a}+\mathbf{b}\| \le \|\mathbf{a}\| + \|\mathbf{b}\|$, to separate the contribution of the ADR residual from the path-geometry term. We then take expectations and use the elementary Jensen (equivalently, Cauchy–Schwarz) inequality, $\mathbb{E}\|Z\| \le \sqrt{\mathbb{E}\|Z\|^2}$ for any random vector Z. This yields the following sequence of upper bounds:

$$\begin{aligned}\mathbb{E}\|\mathbf{e}_{\tau_K}\| &\le \mathbb{E}\left\|\int_0^1 e^{\,1-\tau}\,\boldsymbol{\phi}_\tau\,d\tau\right\| + (e-2)\,\|\mathbf{x}_*-\boldsymbol{\epsilon}\|\\ &\le \int_0^1 e^{\,1-\tau}\,\mathbb{E}\|\boldsymbol{\phi}_\tau\|\,d\tau + (e-2)\,\|\mathbf{x}_*-\boldsymbol{\epsilon}\|\\ &\le \int_0^1 e^{\,1-\tau}\,\sqrt{\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2}\,d\tau + (e-2)\,\|\mathbf{x}_*-\boldsymbol{\epsilon}\|.\end{aligned} \tag{49}$$

This establishes an upper bound governed by the integrated ADR residual and a path-geometry constant.

To aid interpretation, we apply an identity transformation to rewrite $\|\mathbf{x}_*-\boldsymbol{\epsilon}\|$ as $\|\boldsymbol{\epsilon}-\mathbf{x}_*\|$, and we can obtain:

$$\mathbb{E}\|\mathbf{e}_{\tau_K}\| \le \int_0^1 e^{\,1-\tau}\,\sqrt{\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2}\,d\tau \;+\; (e-2)\,\|\boldsymbol{\epsilon}-\mathbf{x}_*\|. \tag{50}$$

The final expression in Eq. 50 clarifies the structure of the bound. The first term depends solely on the ADR residual $\boldsymbol{\phi}_\tau$ and hence quantifies the modeling error controlled by the ADR objective. The second term is a model-independent *path-geometry constant*. Consequently, training with the anti-drift target reduces the overall sampling error, including the component arising from exposure bias. Thus, compared with the error bound of the FM target, the ADR objective directly mitigates exposure bias. □

B.3. Relating FM and ADR Residuals

Recall the residuals introduced in Sec. 4.1:

$$\mathbf{r}_\tau := \mathbf{v}_\theta(\mathbf{x}_\tau, \tau) - (\boldsymbol{\epsilon} - \mathbf{x}_*), \tag{51}$$

$$\boldsymbol{\delta}_{\text{exp},\tau} := \mathbf{v}_\theta(\hat{\mathbf{x}}_\tau, \tau) - \mathbf{v}_\theta(\mathbf{x}_\tau, \tau), \tag{52}$$

$$\boldsymbol{\phi}_\tau := \mathbf{v}_\theta(\hat{\mathbf{x}}_\tau, \tau) - (\hat{\mathbf{x}}_\tau - \mathbf{x}_*). \tag{53}$$

We now establish a structural identity linking the ground truth velocity residual $\mathbf{r}_\tau$, the exposure-bias term $\boldsymbol{\delta}_{\text{exp},\tau}$, and the ADR residual $\boldsymbol{\phi}_\tau$.

Lemma B.1 (Coupling between FM and ADR residuals)**.** *For any $\tau \in [0, 1]$, the residuals satisfy the exact identity:*

$$\boldsymbol{\phi}_\tau = \mathbf{r}_\tau + \boldsymbol{\delta}_{\text{exp},\tau} + (\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau). \tag{54}$$

Proof of Lemma.B.1. By definition:

$$\boldsymbol{\phi}_\tau = \mathbf{v}_\theta(\hat{\mathbf{x}}_\tau, \tau) - (\hat{\mathbf{x}}_\tau - \mathbf{x}_*). \tag{55}$$

We first insert $\mathbf{v}_\theta(\mathbf{x}_\tau, \tau)$ into the expression Eq. 55 using the identity of Eq. 52. Then, we subtract and subsequently add $(\boldsymbol{\epsilon} - \mathbf{x}_*)$ to isolate the ground truth velocity residual and exposure-bias terms. Applying these two steps yields:

$$\begin{aligned}
\boldsymbol{\phi}_\tau &= \big[\mathbf{v}_\theta(\mathbf{x}_\tau, \tau) + \boldsymbol{\delta}_{\text{exp},\tau}\big] - (\hat{\mathbf{x}}_\tau - \mathbf{x}_*) \\
&= \underbrace{\big[\mathbf{v}_\theta(\mathbf{x}_\tau, \tau) - (\boldsymbol{\epsilon} - \mathbf{x}_*)\big]}_{\mathbf{r}_\tau} + \boldsymbol{\delta}_{\text{exp},\tau} \\
&\quad + \big[(\boldsymbol{\epsilon} - \mathbf{x}_*) - (\hat{\mathbf{x}}_\tau - \mathbf{x}_*)\big] \\
&= \mathbf{r}_\tau + \boldsymbol{\delta}_{\text{exp},\tau} + (\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau),
\end{aligned} \tag{56}$$

which proves Eq. 54. □

This identity shows that the ADR residual $\boldsymbol{\phi}_\tau$ can be viewed as a re-parameterization of the ground truth velocity residual $\mathbf{r}_\tau$ and the exposure bias $\boldsymbol{\delta}_{\text{exp},\tau}$, plus the purely geometric offset $(\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau)$ along the denoising trajectory.

Corollary B.2 (Order-equivalence of FM and ADR bounds)**.** *Let B_{FM} and B_{ADR} denote the error bounds in Propositions 4.1 and 4.3, respectively, i.e.,*

$$B_{\text{FM}} := \int_0^1 \sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2}\, d\tau + \int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau}\|^2}\, d\tau, \tag{57}$$

$$B_{\text{ADR}} := \int_0^1 e^{1-\tau} \sqrt{\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2}\, d\tau + (e-2)\,\|\boldsymbol{\epsilon} - \mathbf{x}_*\|. \tag{58}$$

Then there exist absolute constants $C_1, C_2 > 0$, independent of $\mathbf{v}_\theta$, such that:

$$B_{\text{ADR}} \;\le\; C_1\, B_{\text{FM}} + C_2\, \|\boldsymbol{\epsilon} - \mathbf{x}_*\|. \tag{59}$$

Proof of Corollary. B.2. From Lemma B.1, the vector $\boldsymbol{\phi}_\tau$ can be decomposed into the sum of three terms. Applying the standard quadratic inequality for vectors, $\|a+b+c\|^2 \le 3(\|a\|^2 + \|b\|^2 + \|c\|^2)$, we obtain:

$$\|\boldsymbol{\phi}_\tau\|^2 \le 3\Big(\|\mathbf{r}_\tau\|^2 + \|\boldsymbol{\delta}_{\text{exp},\tau}\|^2 + \|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2\Big). \tag{60}$$

Taking expectations on both sides and using the linearity of expectation give:

$$\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2 \le 3\Big(\mathbb{E}\|\mathbf{r}_\tau\|^2 + \mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau}\|^2 + \mathbb{E}\|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2\Big). \tag{61}$$

We then take the square root of both sides. Using multiplicativity of the square root over nonnegative scalars, i.e., $\sqrt{ab} = \sqrt{a}\sqrt{b}$ for $a \ge 0$, yields:

$$\sqrt{\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2} \le \sqrt{3}\, \sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2 + \mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau}\|^2 + \mathbb{E}\|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2}\,. \tag{62}$$

Finally, we bound the square root of the sum by applying the subadditivity (i.e., concavity) of the square root function for nonnegative arguments, $\sqrt{u+v+w} \le \sqrt{u} + \sqrt{v} + \sqrt{w}$, $(u, v, w \ge 0)$, which follows directly from $(\sqrt{u}+\sqrt{v}+\sqrt{w})^2 \ge u + v + w$. This yields the final bound:

$$\sqrt{\mathbb{E}\|\boldsymbol{\phi}_\tau\|^2} \le \sqrt{3}\Big(\sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2} + \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau}\|^2} + \sqrt{\mathbb{E}\|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2}\Big), \tag{63}$$

Substituting Eq. 63 into Eq. 58 and using the elementary bound $e^{1-\tau} \le e$ for all $\tau \in [0, 1]$, we obtain:

$$\begin{aligned}
B_{\text{ADR}} &\le e\sqrt{3} \int_0^1 \big(\sqrt{\mathbb{E}\|\mathbf{r}_\tau\|^2} + \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\text{exp},\tau}\|^2}\big) \\
&\quad + e\sqrt{3} \int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2}\, d\tau + (e-2)\|\boldsymbol{\epsilon} - \mathbf{x}_*\| \\
&= e\sqrt{3}\, B_{\text{FM}} + e\sqrt{3} \int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2}\, d\tau \\
&\quad + (e-2)\|\boldsymbol{\epsilon} - \mathbf{x}_*\|.
\end{aligned} \tag{64}$$

We next bound the geometric contribution. Under the forward interpolation $\mathbf{x}_\tau = (1-\tau)\boldsymbol{\epsilon} + \tau\mathbf{x}_*$ and the biased state $\hat{\mathbf{x}}_\tau = \mathbf{x}_\tau + \mathbf{e}_\tau$, we have the exact algebraic decomposition:

$$\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau = (\boldsymbol{\epsilon} - \mathbf{x}_\tau) - \mathbf{e}_\tau = \tau(\boldsymbol{\epsilon} - \mathbf{x}_*) - \mathbf{e}_\tau. \tag{65}$$

Taking expectations in Eq. 65 and applying the triangle inequality for norms $\|u+v\| \le \|u\| + \|v\|$ yields the bound:

$$\sqrt{\mathbb{E}\|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2} \le \tau\|\boldsymbol{\epsilon} - \mathbf{x}_*\| + \sqrt{\mathbb{E}\|\mathbf{e}_\tau\|^2}. \tag{66}$$

Integrating Eq. 66 over $\tau \in [0,1]$ gives:

$$\int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2}\, d\tau \le \frac{1}{2}\|\boldsymbol{\epsilon} - \mathbf{x}_*\| + \int_0^1 \sqrt{\mathbb{E}\|\mathbf{e}_\tau\|^2}\, d\tau. \tag{67}$$

To bound the remaining term, we repeat the derivation of Proposition 4.1, but integrate the recursion only up to time τ rather than 1. This yields, for every $\tau \in [0,1]$,

$$\sqrt{\mathbb{E}\|\mathbf{e}_\tau\|^2} \le \int_0^\tau \sqrt{\mathbb{E}\|\mathbf{r}_s\|^2}\, ds + \int_0^\tau \sqrt{\mathbb{E}\|\boldsymbol{\delta}_{\exp,s}\|^2}\, ds \le B_{\mathrm{FM}}. \tag{68}$$

Therefore,

$$\int_0^1 \sqrt{\mathbb{E}\|\mathbf{e}_\tau\|^2}\, d\tau \le \int_0^1 B_{\mathrm{FM}}\, d\tau = B_{\mathrm{FM}}. \tag{69}$$

Combining Eq. 67 and Eq. 69, we obtain:

$$\int_0^1 \sqrt{\mathbb{E}\|\boldsymbol{\epsilon} - \hat{\mathbf{x}}_\tau\|^2}\, d\tau \le B_{\mathrm{FM}} + \tfrac{1}{2}\|\boldsymbol{\epsilon} - \mathbf{x}_*\|. \tag{70}$$

Plugging this estimate Eq. 70 into Eq. 64 shows that B_{ADR} is bounded by a constant multiple of B_{FM} plus a constant multiple of $\|\boldsymbol{\epsilon} - \mathbf{x}_*\|$. Thus there exist absolute constants $C_1, C_2 > 0$ such that:

$$\begin{aligned} B_{\mathrm{ADR}} &\le e\sqrt{3}\, B_{\mathrm{FM}} + e\sqrt{3}\Big(B_{\mathrm{FM}} + \tfrac{1}{2}\|\boldsymbol{\epsilon} - \mathbf{x}_*\|\Big) \\ &\quad + (e-2)\|\boldsymbol{\epsilon} - \mathbf{x}_*\| \\ &= C_1\, B_{\mathrm{FM}} + C_2\, \|\boldsymbol{\epsilon} - \mathbf{x}_*\|, \end{aligned} \tag{71}$$

where $C_1 = 2e\sqrt{3}, C_2 = \frac{e\sqrt{3}}{2} + (e-2)$, and e denotes Euler's number. It proves Eq. 59. □

Remark B.3. When B_{ADR} and B_{FM} are not of the same magnitude order, it becomes unclear how to compare their effectiveness in measuring the sampling error. Corollary B.2 shows that, even in the worst case, the B_{ADR} remains of the same order as the original B_{FM}. This ensures that B_{ADR} provides a estimate of sampling error with reliability comparable to B_{FM}. Unlike B_{FM}, however, B_{ADR} also includes the exposure bias term that FM training does not minimize. As a result, the ADR objective offers more complete control over the sources of sampling error and can more effectively alleviate exposure bias.

C. More Experiments

C.1. Multi-Step Scheduled Sampling Training of ADR

In our main experiments, we show that one-step inference training already achieves strong improvements. We further investigate whether extending the training procedure to multi-step inference provides additional gains. However, increasing the number of simulated inference steps also increases computational cost. Therefore, we extend the one-step setting to two- and three-step variants.

Table 7. **The effect of multi-step scheduled sampling training of ADR under 50-NFE generation.** Unconditional generation on CIFAR-10.

Method	Steps	Timestep Sampling	Objective	CIFAR-10(uncond)	
				FID↓	IS↑
SiT [30]	/	/	$\mathcal{L}_{FM}$	17.41	7.82
ReflexFlow w/o FC	1	Random	Regularizer	<u>15.92</u>	7.83
ReflexFlow w/o FC	2	Random	Regularizer	21.89	**8.10**
ReflexFlow w/o FC	2	Equal	Regularizer	**15.33**	<u>8.04</u>
ReflexFlow w/o FC	3	Random	Regularizer	148.95	3.76
ReflexFlow w/o FC	3	Equal	Regularizer	32.20	7.30

In the **Random** timestep sampling strategy, we first randomly sample a timestep $t_0 \in [0,1]$, then sample t_1 randomly within $[0, t_0)$, and continue recursively for more steps. In contrast, after sampling t_0 timestep, the **Equal** strategy partitions the sub-interval $[0, t_0)$ into n equal intervals for n-step simulation, and each subsequent timestep is sampled randomly within its designated sub-interval. This encourages stabler timestep sampling compared to the fully random scheme.

As shown in the third and fourth rows of Tab. 7, two-step training improves performance over the one-step case. However, extending to three steps leads to degradation (the fifth and sixth rows). A plausible explanation is that increasing the number of simulated inference steps during training raises the modeling burden, making it harder for the network to learn stable and consistent transitions. The Equal strategy consistently performs better than Random, demonstrating its improved stability. Notably, when using Random sampling, three-step training collapses, indicating that timestep selection plays a critical role in stabilizing multi-step scheduled sampling. The Equal strategy achieves more reliable behavior under multi-step scheduled sampling training.

Table 8. **Extended-Training stability and effectiveness of ReflexFlow.** Unconditional generation on CIFAR-10. All entries are reported using 50-NFE samplings. Models are trained from 200k to 1.5M iterations with SiT-S/8.

Method	Params	iterations	CIFAR10(uncond)	
			FID↓	IS↑
SiT-S/8	34M	200k	68.35	4.35
ReflexFlow(ours)	34M	200k	**51.04**	**5.44**
SiT-S/8	34M	500k	60.70	4.86
ReflexFlow(ours)	34M	500k	**41.22**	**6.21**
SiT-S/8	34M	1M	51.88	5.23
ReflexFlow(ours)	34M	1M	**34.62**	**6.62**
SiT-S/8	34M	1.5M	49.53	5.44
ReflexFlow(ours)	34M	1.5M	**30.12**	**6.69**

C.2. Extended-Training Stability and Effectiveness of ReflexFlow

Due to computational constraints, we train SiT-S/8 on CIFAR-10 to evaluate the stability and effectiveness of ReflexFlow over 1.5M training iterations. As shown in Tab. 8, we observe that ADR consistently provides improved performance, while ReflexFlow achieves the best overall results.

C.3. Multi-NFE on ImageNet-256 with CFG

Table 9. **Multi-NFE on ImageNet-256 analysis with cfg = 1.5.** Results are reported using a SiT-B/4 model trained for 500k iterations under 50-NFE conditional generation.

Steps	SiT		IP		SDSS		MDSS		Ours	
	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑	FID↓	IS↑
30	29.08	55.64	26.84	**59.14**	28.10	58.24	29.47	56.61	**26.00**	56.43
50	28.22	55.65	25.98	**59.20**	27.03	58.69	28.44	56.89	**25.07**	58.94
100	27.81	55.51	25.50	59.04	26.49	58.66	27.84	56.93	**24.86**	**59.39**
250	27.68	55.15	25.36	58.59	26.30	58.38	27.63	56.63	**24.80**	**59.38**
500	27.68	54.98	25.36	58.32	26.30	58.14	27.62	56.57	**24.83**	**59.36**

In addition to evaluating multi-step sampling under the without-CFG setting, we further test our method with Classifier-Free Guidance (CFG) set to 1.5. As illustrated in Tab. 9, the observations are consistent with the without-CFG case. With CFG enabled, our method remains stable across different NFEs and achieves the lowest FID throughout. Although our IS score is not always the best at very small NFEs, it improves as the number of sampling steps increases. This trend reflects the reduced exposure bias achieved by our approach, which helps maintain performance during extended sampling. In contrast, the strong baseline IP obtains competitive results at small NFEs but degrades continuously as the sampling steps grow, indicating that ReflexFlow is more effective at alleviating exposure bias.

D. More Related Work

Flow Rectification. Recent advances show that directly modifying flow trajectories can effectively enhance generation quality and stability. Several works aim to accelerate sampling by redesigning the flow operators. Shortcut Models [13] introduce shortcut paths to enable one-step generation, Mean Flows [14] approximate the expected flow for deterministic mapping, and SplitMeanFlow [15] enforces interval-wise consistency to stabilize few-step modeling. To reduce sampling drift, Consistency Models [45] and Consistent Diffusion [6] learn drift-free mappings via self-consistency, while Consistency Trajectory Models [19, 59] constrain entire probability flow ODE trajectories. In contrast, CurveFlow [29] leverages curvature to guide smoother flows, FlowEdit [22] modifies flow fields for text-based editing, and Align-Your-Flow [39] aligns student–teacher flow maps for scalable distillation. However, these methods primarily focus on acceleration, stability, or control, rather than explicitly addressing the exposure bias caused by the mismatch between training and inference in FM.

E. Additional Experimental Details

We follow the hyperparameters of Mean Flows [14] for ImageNet-256, and those of IP [31] for CIFAR-10 and CelebA-64, as specified in Tab. 10. For ImageNet-256, we follow the standard VAE tokenizer to obtain latent representations of size $32 \times 32 \times 4$, which serve as the model input. The backbone follows SiT [30], built on ViT [10] with adaLN-Zero [35] for conditioning. For CIFAR-10, the model operates directly on $32 \times 32 \times 3$ inputs. We evaluate both conditional and unconditional generation. The conditional setup follows the same configuration as ImageNet-256. For the unconditional setting, we disable class conditioning by assigning all samples to a single class. For CelebA-64, the model directly takes and operates on $64 \times 64 \times 3$ pixel inputs. We perform only unconditional generation, again using a single-class setup. The backbone remains identical to the SiT architecture used in the other datasets. We use Pytorch 2.1 [34], adam [1] and trained all the models on Ascend 910B NPUs (64G memory). For statistics that depend only on a single timestep (e.g., Fig. 2, Fig. 8 and Fig. 9), we compute the metric for every sample at each timestep and report the average. In contrast, the metrics like R_{LFDV} of Exposure Bias and R_{LFDL} of FC loss in Fig. 3 require exposure bias, which depends on differences between two timesteps. For these metrics, the value at target timestep t is obtained by aggregating exposure bias over all intermediate timesteps in $[0, t)$ and then averaging across samples. In all statistical experiments, we discretize the time interval $[0, 1]$ into 50 uniform timesteps. Each reported value corresponds to the mean over all samples at the designated timestep, with exposure bias related metrics additionally aggregating over all preceding intermediate steps.

F. Baseline Implementation Details

SiT [30] Details. We directly follow the hyperparameters and methods of SiT to train the models and reproduce the results.

IP [31] Details. We adopt the default perturbation scale of 0.1. The core idea of IP is to add input perturbations to simulate inference-time errors. Following the original design, we inject an additional noise term of magnitude 0.1 into the input to mimic such errors and improve robustness to inference drift. This modification requires only a single-line code change to the original SiT implementation.

SDSS [37] Details. The method highlights its most effective strategy to align predictions with the ground truth after simulating one inference step. This alignment improves robustness to drift accumulated during sampling. Following

Table 10. **Configurations on ImageNet-256, CelebA-64 and CIFAR-10.**

configs	**SiT-B/4**	**SiT-M/2**	**SiT-XL/2**	**SiT-XL/2+**	**SiT-B/4**	**SiT-B/4**
Dataset	ImageNet-256	ImageNet-256	ImageNet-256	ImageNet-256	CelebA-64	CIFAR-10
Params (M)	131	308	675	675	131	131
Depth	12	16	28	28	12	12
Hidden dim	768	1024	1152	1152	768	768
Heads	12	16	16	16	12	12
Patch size	4×4	2×2	2×2	2×2	4×4	4×4
Training iterations	500k	500k	500k	700k	250k	250k
Global batch size	256	256	256	256	256	128
Dropout	0.0					
Optimizer	Adam [1]					
Lr schedule	constant					
Learning rate	0.0001					
Adam $(\beta_{\text{Adam},1}, \beta_{\text{Adam},2})$	(0.9, 0.999)					
Weight decay	0.0					
Gradient clip	0.2					
Using NPUs	4	8	32	32	4	1
Second per iterations	0.55	0.57	0.90	0.90	0.45	0.40
α (exposure bias coeff.)	1					
β_1 (ADR loss coeff.)	10					
β_2 (FC loss coeff.)	1					

the paper, we implement SDSS within the SiT backbone under the Flow Matching framework. We randomly sample two timesteps $t_0, t_1 \in [0, 1]$, $t_0 > t_1$, and infer one step from t_0 to t_1 per iteration, obtain the biased velocity at t_1, and then align it with the ground-truth target $\boldsymbol{\epsilon} - \mathbf{x}_*$.

MDSS [37] Details. The paper reports that four training-time inference steps offer the best trade-off, and we therefore adopt the same setup. We randomly sample two timesteps $t_0, t_1 \in [0, 1]$, with $t_0 > t_1$, and perform one-step inference to reach t_1. We then directly accumulate the other three additional steps without model prediction to simulate four-step inference, and finally align the final prediction with the ground truth as specified by MDSS.

G. More Qualitative Comparison on ImageNet-256 and CelebA-64

In this section, we qualitatively compare ReflexFlow with several baselines (SiT, IP, SDSS, and MDSS). For a fair comparison, we use the same random seed and initialize sampling with the same noise across all models. As shown in Fig. 10 and Fig. 11, images generated by ReflexFlow are generally comparable to or better than those produced by the baselines.

In Fig. 10, baseline models often fail to render key semantic structures, such as the bird's feet and beak, the background trees, or the fine shading on the mushroom, leading to overexposed or incomplete details (second, third, and fifth rows). In contrast, ReflexFlow generates sharper boundaries, more coherent textures and more faithfully reconstructed object parts.

Similarly, in Fig. 11, baselines struggle to produce clean facial regions, leaving dark patches on the forehead, blurred or inconsistent hair structures, or incorrect separation between the neck and clothing. They also exhibit artifacts around the face and fail to maintain consistent geometry between the head and neck. From the first to the seventh row, ReflexFlow produces cleaner facial details, more accurate hair shapes, clearer background-to-foreground transitions, and reduced color artifacts, resulting in overall higher visual fidelity.

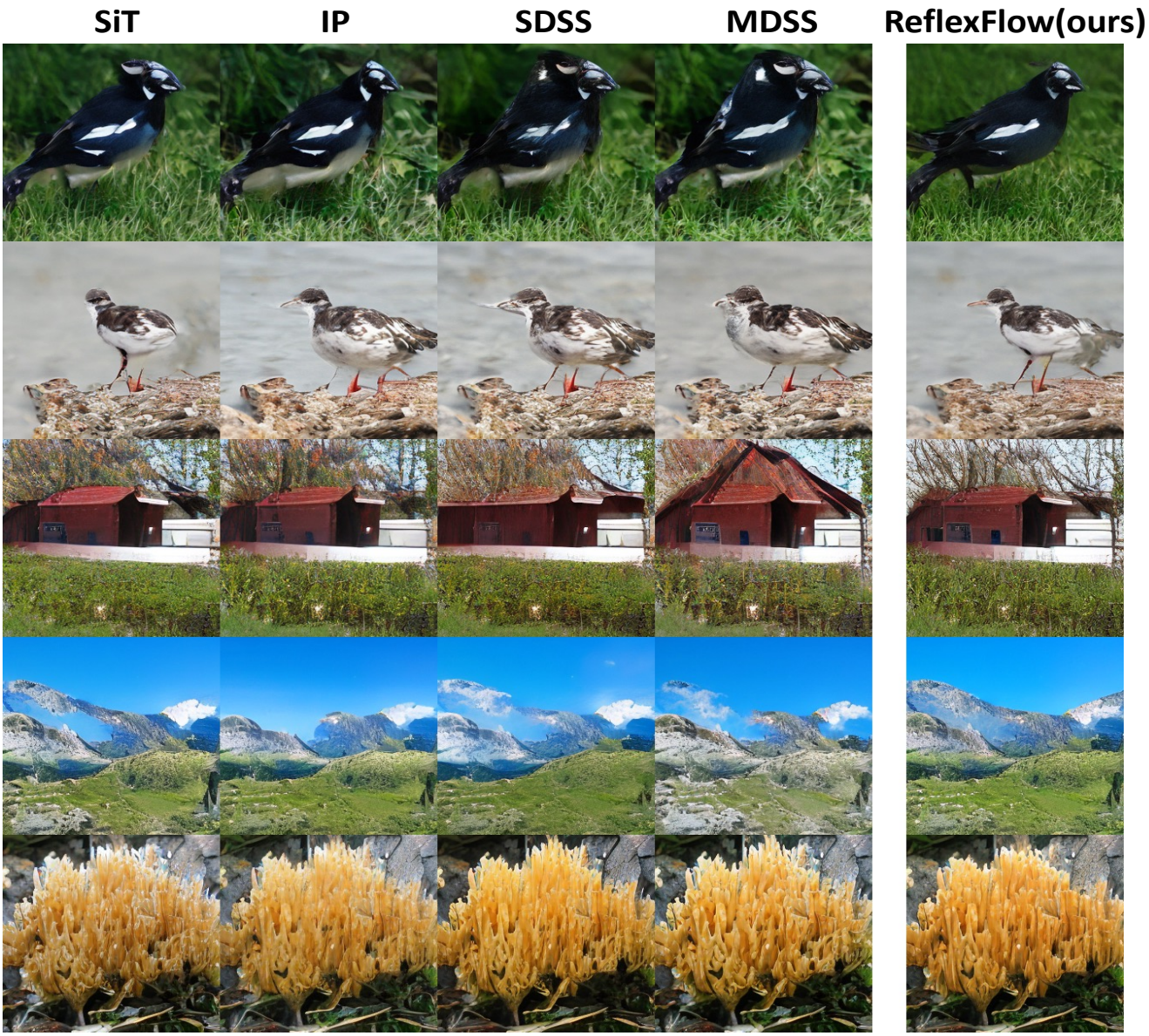

Figure 10. **Qualitative comparison across methods.** On ImageNet-256 dataset under 50-NFE generation with cfg=1.5, our approach produces consistently better visual quality than all baseline methods.

H. Generated Samples on ImageNet-256 under 50-NFE.

In this section, we present images directly sampled from ReflexFlow to qualitatively evaluate the generative capability of model, as shown in Fig. 12. We use our best-performing ReflexFlow-XL/2+ model and sample with 50 NFEs and CFG=4 on ImageNet-256. Across both static scenes and animal categories, the model produces high-quality samples with fine-grained details. In many cases, ReflexFlow generates realistic animal characteristics, including detailed fur textures, eye reflections, and subtle motion cues. It also captures dynamic scenes, such as volcanic eruptions, with rich structural and textural fidelity.

I. More Frequency-Aware Visualization on ImageNet-256

In this section, we provide detailed examples illustrating that exposure bias is naturally aware of low-frequency regions in samples. We use a SiT-B/4 model trained for 500k iterations and compute exposure bias under HFT for different samples. We then directly visualize the resulting exposure bias maps, as shown in Fig. 13.

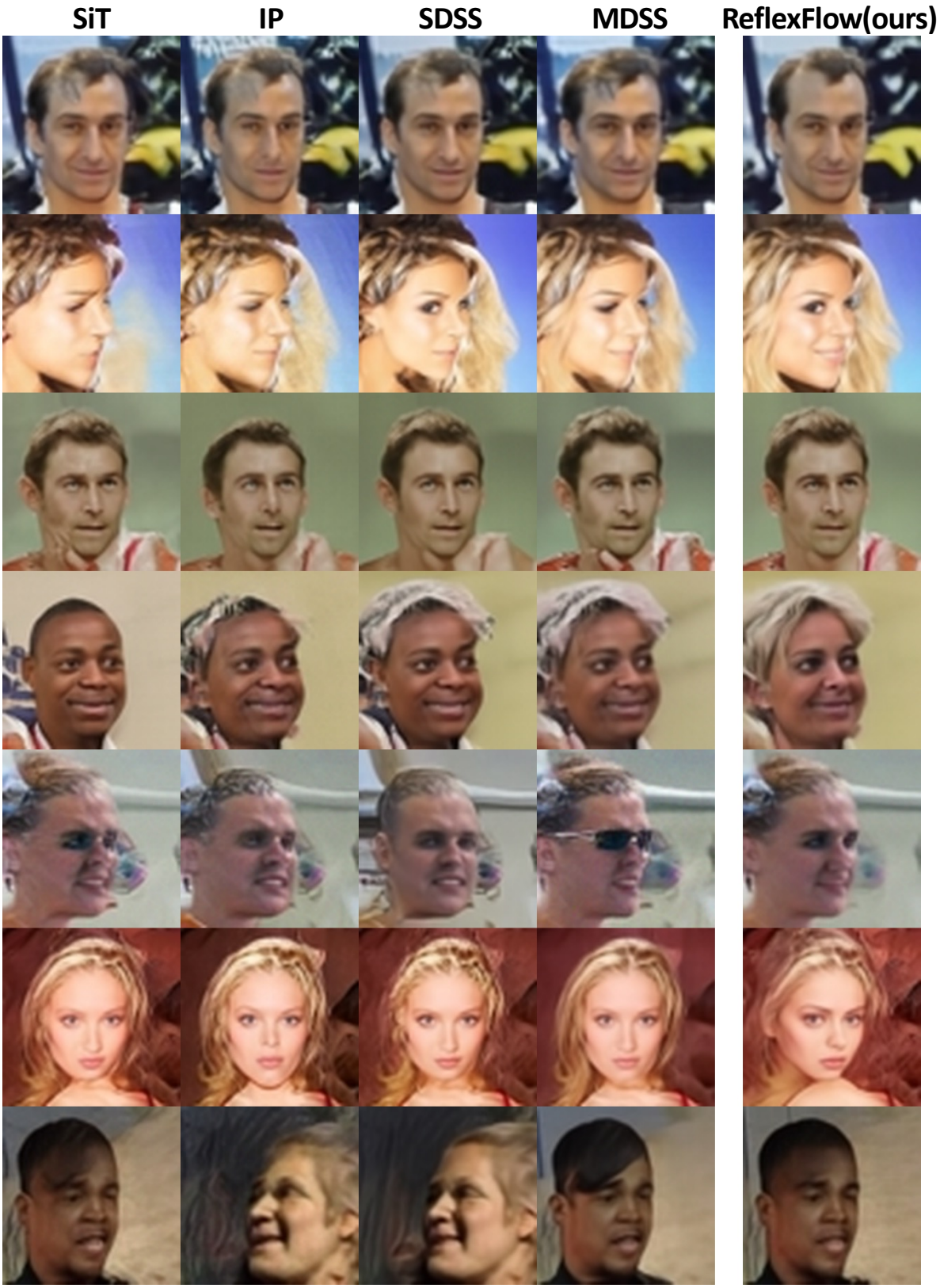

Figure 11. **Qualitative comparison across methods.** On CelebA-64 dataset unconditional generation under 50-NFE sampling, our approach produces consistently better visual quality than all baseline methods.

Figure 12. **ImageNet-256 qualitative results of ReflexFlow with SiT-XL/2+ with cfg=4.** The samples are generated using 50 sampling steps.

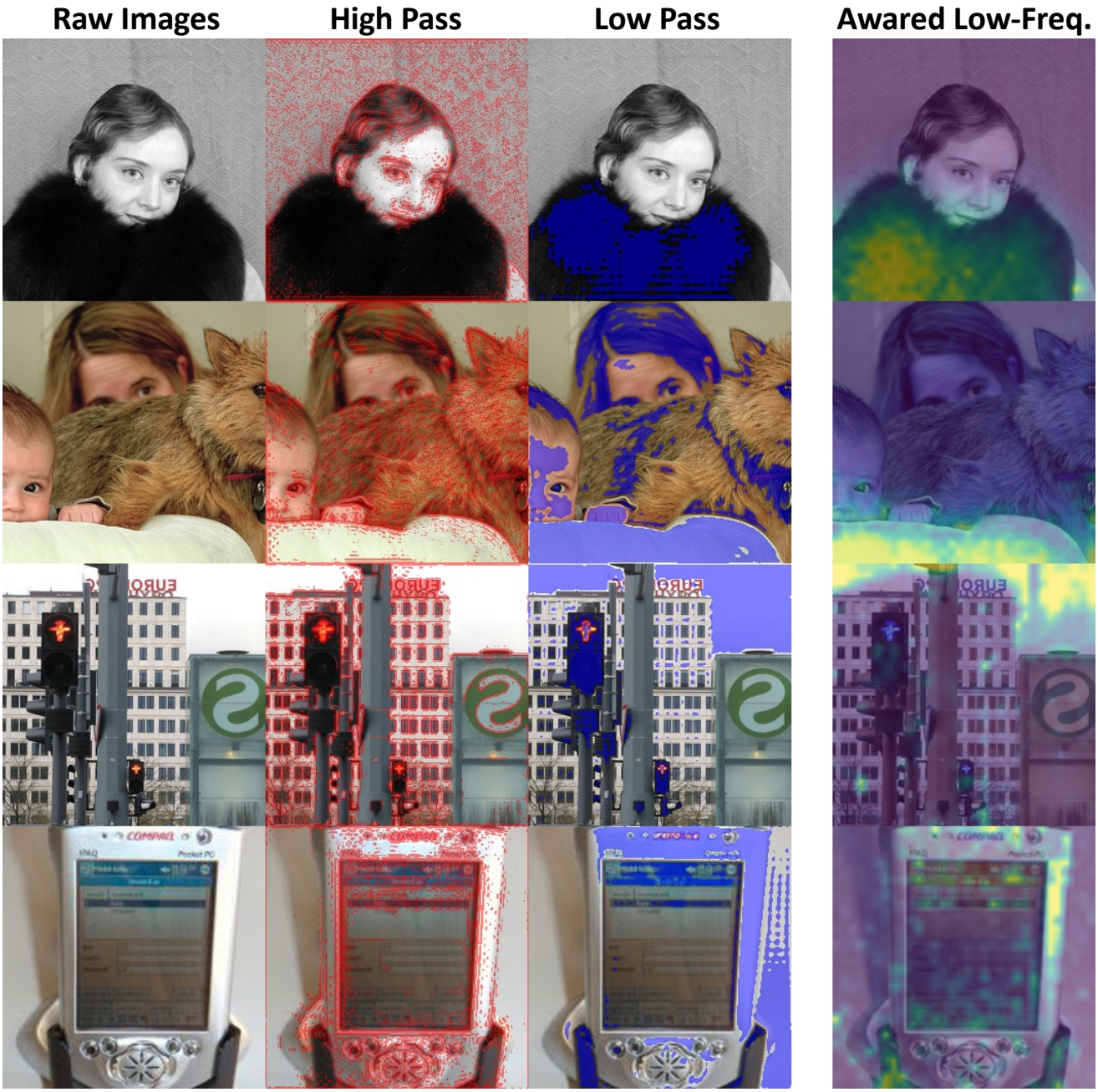

Figure 13. **Exposure bias highlights low-frequency structures in images.** We analyze the frequency components of raw images as references and observe that exposure bias consistently aligns with low-pass regions. These examples are from ImageNet-256.

The leftmost column of Fig. 13 shows the original images. Applying standard high- and low-pass filters to these images yields their high-frequency and low-frequency components (second and third columns). The last column presents the visualization of the raw exposure bias. Across the first four rows, exposure bias consistently highlights low-frequency regions, such as the dark clothing wrapped around the woman, the sofa in front of the baby and the cat, the sky regions in outdoor scenes, and the smooth surface of the phone casing.

These results demonstrate that exposure bias inherently attends to semantically meaningful low-frequency pixels under HFT.

References

[1] Kingma DP Ba J Adam et al. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412(6), 2014. 18, 19

[2] Michael Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR*, 2023. 1

[3] Chengyu Bai, Yuming Li, Zhongyu Zhao, Jintao Chen, Peidong Jia, Qi She, Ming Lu, and Shanghang Zhang. Fastinit: Fast noise initialization for temporally consistent video generation. *arXiv preprint arXiv:2506.16119*, 2025. 6

[4] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *NeurIPS*, 28, 2015. 2

[5] Ziyang Chen, Prem Seetharaman, Bryan Russell, Oriol Nieto, David Bourgin, Andrew Owens, and Justin Salamon. Video-guided foley sound generation with multimodal controls. In *CVPR*, pages 18770–18781, 2025. 1

[6] Giannis Daras, Yuval Dagan, Alex Dimakis, and Constantinos Daskalakis. Consistent diffusion models: Mitigating sampling drift by learning to be consistent. *NeurIPS*, 36: 42038–42063, 2023. 18

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 6

[8] Yuntian Deng, Noriyuki Kojima, and Alexander M Rush. Markup-to-image diffusion models with scheduled sampling. In *ICLR*, 2022. 2

[9] Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025. 1

[10] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 18

[11] Martin Nicolas Everaert, Athanasios Fitsios, Marco Bocchio, Sami Arpa, Sabine Süsstrunk, and Radhakrishna Achanta. Exploiting the signal-leak bias in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4025–4034, 2024. 2, 3

[12] Haopeng Fang, Di Qiu, Binjie Mao, Pengfei Yan, and He Tang. Motioncharacter: Identity-preserving and motion controllable human video generation. *arXiv preprint arXiv:2411.18281*, 2024. 3, 6

[13] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *ICLR*, 2024. 18

[14] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025. 7, 18

[15] Yi Guo, Wei Wang, Zhihang Yuan, Rong Cao, Kuan Chen, Zhengyang Chen, Yuanyuan Huo, Yang Zhang, Yuping Wang, Shouda Liu, et al. Splitmeanflow: Interval splitting consistency in few-step generative modeling. *arXiv preprint arXiv:2507.16884*, 2025. 18

[16] Moayed Haji-Ali, Willi Menapace, Ivan Skorokhodov, Arpit Sahni, Sergey Tulyakov, Vicente Ordonez, and Aliaksandr Siarohin. Improving progressive generation with decomposable flow matching. *arXiv preprint arXiv:2506.19839*, 2025. 3

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 7

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1

[19] Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. In *ICLR*, 2024. 18

[20] Kwanyoung Kim and Sanghyun Kim. Model already knows the best noise: Bayesian active noise selection via attention in video diffusion model. *arXiv preprint arXiv:2505.17561*, 2025. 6

[21] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 7

[22] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. Flowedit: Inversion-free text-based editing using pre-trained flow models. In *ICCV*, pages 19721–19730, 2025. 18

[23] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 1

[24] Mingxiao Li, Tingyu Qu, Ruicong Yao, Wei Sun, and Marie-Francine Moens. Alleviating exposure bias in diffusion models through sampling with shifted time steps. In *ICLR*, 2024. 2, 3

[25] Yangming Li and Mihaela van der Schaar. On error propagation of diffusion models. In *ICLR*, 2024. 2

[26] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 1

[27] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023. 1

[28] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015. 7

[29] Yan Luo, Drake Du, Hao Huang, Yi Fang, and Mengyu Wang. Curveflow: Curvature-guided flow matching for image generation. *arXiv preprint arXiv:2508.15093*, 2025. 18

[30] Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *ECCV*, pages 23–40. Springer, 2024. 1, 7, 9, 10, 17, 18

[31] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. *arXiv preprint arXiv:2301.11706*, 2023. 1, 2, 7, 18

[32] Mang Ning, Mingxiao Li, Jianlin Su, Albert Ali Salah, and Itir Onal Ertugrul. Elucidating the exposure bias in diffusion models. In *ICLR*, 2024. 2, 3

[33] Anton Obukhov, Maximilian Seitzer, Po-Wei Wu, Semen Zhydenko, Jonathan Kyl, and Elvis Yu-Jing Lin. High-fidelity performance metrics for generative models in pytorch, 2020. Version: 0.3.0, DOI: 10.5281/zenodo.4957738. 7

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 18

[35] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pages 4195–4205, 2023. 18

[36] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015. 2

[37] Zhiyao Ren, Yibing Zhan, Liang Ding, Gaoang Wang, Chaoyue Wang, Zhongyi Fan, and Dacheng Tao. Multi-step denoising scheduled sampling: Towards alleviating exposure bias for diffusion models. In *AAAI*, pages 4667–4675, 2024. 1, 2, 3, 7, 18, 19

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 7

[39] Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. *arXiv preprint arXiv:2506.14603*, 2025. 18

[40] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29, 2016. 7

[41] Florian Schmidt. Generalization in generation: A closer look at exposure bias. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 157–167. Association for Computational Linguistics, 2019. 2

[42] Luigi Sigillo, Shengfeng He, and Danilo Comminiello. Latent wavelet diffusion: Enabling 4k image synthesis for free. *arXiv preprint arXiv:2506.00433*, 2025. 3, 6

[43] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning(ICML)*, pages 2256–2265. pmlr, 2015. 1

[44] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019. 1

[45] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 32211–32252, 2023. 18

[46] Meituan LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, et al. Longcat-video technical report. *arXiv preprint arXiv:2510.22200*, 2025. 1

[47] Arun Venkatraman, Byron Boots, Martial Hebert, and J Andrew Bagnell. Data as demonstrator with applications to system identification. In *ALR Workshop, NIPS*, 2014. 2

[48] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1

[49] Chaojun Wang and Rico Sennrich. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, 2020. 2

[50] Shijie Wang, Samaneh Azadi, Rohit Girdhar, Saketh Rambhatla, Chen Sun, and Xi Yin. Motif: Making text count in image animation with motion focal loss. In *CVPR*, pages 7773–7783, 2025. 3, 6

[51] Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. In *ECCV*, pages 378–394. Springer, 2024. 6

[52] Meng Yu and Kun Zhan. Frequency regulation for exposure bias mitigation in diffusion models. In *ACM MM*, pages 10370–10378, 2025. 2, 3

[53] YAO Yuzhe, Jun Chen, Zeyi Huang, Haonan Lin, Mengmeng Wang, Guang Dai, and Jingdong Wang. Manifold constraint reduces exposure bias in accelerated diffusion sampling. In *ICLR*, 2025. 3

[54] Guiyu Zhang, Chen Shi, Zijian Jiang, Xunzhi Xiang, Jingjing Qian, Shaoshuai Shi, and Li Jiang. Proteus-id: Id-consistent and motion-coherent video customization. *arXiv preprint arXiv:2506.23729*, 2025. 3, 6

[55] Junyu Zhang, Daochang Liu, Eunbyung Park, Shichao Zhang, and Chang Xu. Anti-exposure bias in diffusion models. In *ICLR*, 2025. 2

[56] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, 2019. 2

[57] Yizhe Zhang, Jiatao Gu, Zhuofeng Wu, Shuangfei Zhai, Joshua Susskind, and Navdeep Jaitly. Planner: Generating diversified paragraph via latent language diffusion model. *NeurIPS*, 36:80178–80190, 2023. 2

[58] Min Zhao, Hongzhou Zhu, Chendong Xiang, Kaiwen Zheng, Chongxuan Li, and Jun Zhu. Identifying and solving conditional image leakage in image-to-video diffusion model. *NeurIPS*, 37:30300–30326, 2024. 6

[59] Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping. *arXiv preprint arXiv:2402.19159*, 2024. 18